

ESTUDIO DE INCLUSIÓN DEL USO DE SOFTWARE LIBRE PARA LA
ENSEÑANZA DE LENGUAJE DE MODELAMIENTO UNIFICADO EN
INGENIERÍA DE SOFTWARE

CHRISTIAM ALEJANDRO NIÑO PEÑA

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE POSGRADOS
ESPECIALIZACIÓN EN DOCENCIA UNIVERSITARIA
BOGOTÁ D.C.
2018

ESTUDIO DE INCLUSIÓN DEL USO DE SOFTWARE LIBRE PARA LA
ENSEÑANZA DE LENGUAJE DE MODELAMIENTO UNIFICADO EN
INGENIERÍA DE SOFTWARE

CHRISTIAM ALEJANDRO NIÑO PEÑA

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE ESPECIALISTA EN
DOCENCIA UNIVERSITARIA

DIRECTOR
FREDY EDUARDO DUARTE LÓPEZ

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE POSGRADOS
ESPECIALIZACIÓN EN DOCENCIA UNIVERSITARIA
BOGOTÁ D.C.
2018

Nota de aceptación:

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Universidad Piloto de Colombia para optar al título de Especialista en Docencia Universitaria

Bogotá D.C., 14 de noviembre de 2018

CONTENIDO

	pág.
INTRODUCCIÓN	7
1. JUSTIFICACIÓN	8
2. PROBLEMA	9
3. OBJETIVOS	10
3.1. OBJETIVO GENERAL.....	10
3.2. OBJETIVOS ESPECIFICOS	10
4. MARCOS DE REFERENCIA.....	11
4.1. MARCO HISTÓRICO	11
4.2. MARCO CONCEPTUAL.....	20
4.3. MARCO LEGAL.....	23
5. CARACTERIZACION DE LA POBLACION	26
6. PROCEDER METODOLOGICO.....	27
7. DESARROLLO DE LA PROPUESTA.....	29
8. CONCLUSIONES.....	64
9. RECOMENDACIONES	65
10. BIBLIOGRAFÍA.....	66

LISTA DE TABLAS

	pág.
Tabla 1. Licencias de software libre.....	14
Tabla 2. Edades de los encuestados	31
Tabla 3. Cantidad de genero de los encuestados	32
Tabla 4. Resultados pregunta 9	33
Tabla 5. Resultados pregunta 18	33
Tabla 6. Resultados pregunta 19	34
Tabla 7. Resultados pregunta 20	35
Tabla 8. Resultados pregunta 25	36
Tabla 9. Resultados pregunta 26	36
Tabla 10. Resultados pregunta 1	38
Tabla 11. Resultados pregunta 2	39
Tabla 12. Resultados pregunta 7	40
Tabla 13. Resultados pregunta 11	41
Tabla 14. Resultados pregunta 15	42
Tabla 15. Resultados pregunta 22	43
Tabla 16. Resultados pregunta 3	44
Tabla 17. Resultados pregunta 5	44
Tabla 18. Resultados pregunta 14	45
Tabla 19. Resultados pregunta 16	46
Tabla 20. Resultados pregunta 17	47
Tabla 21. Resultados pregunta 21	47
Tabla 22. Resultados pregunta 23	48
Tabla 23. Resultados pregunta 24	49
Tabla 24. Resultados pregunta 29	50
Tabla 25. Resultados pregunta 4	51
Tabla 26. Resultados pregunta 6	52
Tabla 27. Resultados pregunta 8	53
Tabla 28. Resultados pregunta 10	54
Tabla 29. Resultados pregunta 12	55
Tabla 30. Resultados pregunta 13	55
Tabla 31. Resultados pregunta 27	56
Tabla 32. Resultados pregunta 28	57
Tabla 33. Resultados pregunta 30	58
Tabla 34. Resultados globales	60

LISTA DE FIGURAS

	pág.
Figura 1. Evolución de UML.....	17
Figura 2. Porcentaje de edades de los encuestados	31
Figura 3. Porcentaje de genero.....	32
Figura 4. Porcentajes de resultados de la pregunta 9.....	33
Figura 5. Porcentajes de resultados de la pregunta 18.....	34
Figura 6. Porcentajes de resultados de la pregunta 19.....	35
Figura 7. Porcentajes de resultados de la pregunta 20.....	36
Figura 8. Porcentajes de resultados de la pregunta 25.....	37
Figura 9. Porcentajes de resultados de la pregunta 26.....	37
Figura 10. Porcentajes de resultados de la pregunta 1.....	38
Figura 11. Porcentajes de resultados de la pregunta 2.....	39
Figura 12. Porcentajes de resultados de la pregunta 7.....	40
Figura 13. Porcentajes de resultados de la pregunta 11.....	41
Figura 14. Porcentajes de resultados de la pregunta 15.....	42
Figura 15. Porcentajes de resultados de la pregunta 22.....	43
Figura 16. Porcentajes de resultados de la pregunta 3.....	44
Figura 17. Porcentajes de resultados de la pregunta 5.....	45
Figura 18. Porcentajes de resultados de la pregunta 14.....	45
Figura 19. Porcentajes de resultados de la pregunta 16.....	46
Figura 20. Porcentajes de resultados de la pregunta 17.....	47
Figura 21. Porcentajes de resultados de la pregunta 21.....	48
Figura 22. Porcentajes de resultados de la pregunta 23.....	49
Figura 23. Porcentajes de resultados de la pregunta 9.....	49
Figura 24. Porcentajes de resultados de la pregunta 29.....	50
Figura 25. Porcentajes de resultados de la pregunta 4.....	51
Figura 26. Porcentajes de resultados de la pregunta 6.....	52
Figura 27. Porcentajes de resultados de la pregunta 8.....	53
Figura 28. Porcentajes de resultados de la pregunta 10.....	54
Figura 29. Porcentajes de resultados de la pregunta 12.....	55
Figura 30. Porcentajes de resultados de la pregunta 13.....	56
Figura 31. Porcentajes de resultados de la pregunta 27.....	57
Figura 32. Porcentajes de resultados de la pregunta 28.....	58
Figura 33. Porcentajes de resultados de la pregunta 30.....	59

INTRODUCCIÓN

En el presente documento se abordará la pertinencia en el uso de software libre en los procesos de enseñanza de ingeniería de software, enfocándose en el problema de cuál sería el impacto de usar software libre en la formación de los estudiantes, lo anterior como punto fundamental para una correcta enseñanza del tema de estudio.

Para lograr esto se utiliza un enfoque evaluativo del tipo cualitativo e interpretativo, usando una muestra de estudiantes por medio de una encuesta que ofrecerá elementos importantes para el análisis y posterior respuesta del objetivo general de la investigación.

En un primer tiempo se establecerá un cuestionario de 30 preguntas que permitirán conocer la inclusión de software libre en el aprendizaje, clasificadas según la edad y género de los encuestados, además de incluirse temas más específicos relacionados con la enseñanza y aprendizaje de software libre, puntos como satisfacción, conocimientos, uso y entendimiento.

Para finalizar se analizará y expondrá los resultados encontrados permitiendo responder a la pregunta planteada de la investigación mediante tabulación y representación de los resultados, y de esta forma poder determinar si es o no pertinente el uso de este tipo de software en la enseñanza de ingeniería de software.

1. JUSTIFICACIÓN

En el proceso de ejecución de la ingeniería de software se debe cumplir una serie de pasos en cada una de sus fases que comprenden el análisis, diseño, desarrollo, pruebas, documentación y mantenimiento, estas se diferencian unas de otras en los productos que generan, uno de estos productos son los diagramas que se crean a partir del lenguaje de modelamiento unificado por sus siglas UML, este lenguaje comprende varios elementos que interactúan entre sí para representar el producto que se desea implementar.

La presente investigación busca esclarecer la pertinencia de utilizar software libre en los procesos de enseñanza de ingeniería de software en el ámbito de educación superior, y de esta forma optimizar los modelos pedagógicos utilizados en pro del aprendizaje y apropiación del conocimiento. En específico, la fase de diseño utilizando notación UML para la representación de diagramas.

Teniendo en cuenta lo anterior, los ingenieros de software generan los diagramas requeridos en la fase de diseño y por ende deben cumplir con los estándares que exige la notación UML para documentar con calidad los productos software. Con base a esto se requiere que su formación académica al desarrollar notación UML sea la más adecuada con el fin de cumplir con los estándares de calidad exigidos por la ingeniería de software. Además, es de anotar la importancia del uso del lenguaje UML en la formación de profesionales en Ingeniería de Sistemas tanto para los perfiles profesionales, como contenido de formación académica, este trabajo busca validar la inclusión de este tipo de herramientas en el ámbito de la educación superior y la docencia universitaria y así poder justificar el uso de las mismas en pro del mejoramiento del aprendizaje.

Además, este trabajo hace parte del ejercicio investigativo resultado de la especialización en docencia universitaria, busca fortalecer el proceso formativo propio y del programa académico en temas pedagógicos y didácticos, además de orientarse a nuevas tecnologías y tic en la educación.

2. PROBLEMA

Actualmente, la ingeniería de software aplica metodologías con las que se desarrolla software de calidad, esas metodologías como cascada, de prototipos, incremental, espiral, desarrollo de aplicaciones rápidas o programación extrema, integran fases en todo el proceso de creación de un producto por lo que es fundamental que el ingeniero de software tenga completo dominio de cada una y sobre todo de los elementos que las constituyen.

Una de las fases que tiene gran importancia en el proceso de creación de software es el diseño. Esta fase se realiza creando diagramas que muestran la funcionalidad, interfaz y arquitectura del sistema y que se apoya del lenguaje de modelamiento unificado (UML). Con referencia a lo anterior, la etapa de diseño integra una gran variedad de diagramas como: casos de uso, máquina de estados, clases, objetos, componentes, actividades, secuencia, entre otros; cada uno de estos diagramas exponen la funcionalidad a ser desarrollada en algunos componentes del sistema. Para facilitar la creación de estos diagramas han surgido una gran variedad de aplicaciones propietarias y privativas que permiten al ingeniero de software disponer de herramientas para agilizar y mejorar su realización.

Ahora bien, esta fase de diseño muchas veces no cumple con las exigencias de calidad esperada, a causa del desconocimiento de los ingenieros de software en el uso de componentes UML de forma adecuada, esto ocasiona que las etapas posteriores no tengan el suficiente recurso técnico para cumplir a cabalidad los requerimientos del cliente terminando en productos software inservibles.

A efectos de este problema es fundamental que los ingenieros de software conozcan y usen de forma adecuada las herramientas de diseño requeridas que proveen las aplicaciones especializadas para tal fin, sin embargo, las mejores aplicaciones son propietarias y generan gastos adicionales al presupuesto de las fábricas de software razón por la cual se usa en gran medida software libre.

Dadas las condiciones que anteceden, se requiere que los ingenieros de software tengan una formación completa y clara sobre el diseño de software utilizando aplicativos UML de software libre, lo anterior, para el mejoramiento de los procesos de aprendizaje y aplicación de técnicas que permitan apropiarse del conocimiento, entonces surgen la pregunta: ¿Cuál es el impacto de usar software libre para la enseñanza de UML en la formación de estudiantes de ingeniería de Software?, el presente trabajo permitirá llegar a obtener las respuestas esperadas y demostrarlas.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Evaluar la inclusión del uso de software libre en la enseñanza de lenguaje de modelamiento unificado específicamente en la fase de diseño para ingeniería de software dirigido a instituciones de educación superior.

3.2. OBJETIVOS ESPECIFICOS

- Diseñar y aplicar métodos de recolección de información sobre la inclusión de software libre en la enseñanza de UML en estudiantes de educación superior.
- Presentar la evaluación sobre aspectos determinantes en la inclusión de software libre para la enseñanza de UML en ingeniería de software para la fase de diseño.
- Analizar los resultados obtenidos sobre la inclusión de software libre en la enseñanza de UML.

4. MARCOS DE REFERENCIA

Para el desarrollo de este documento, se presenta el siguiente marco teórico y conceptual de donde se encontraron trabajos de investigación relacionados con el tema que se pretende abordar y que apunta a conocer la pertinencia del uso de software libre en la enseñanza de lenguaje de modelamiento unificado (UML) para la ingeniería de software. En este sentido, se hace prudente realizar, en primera instancia una revisión acerca de aportes de algunos académicos sobre las formas, estilos y prácticas de enseñanza en la educación superior; luego se abordará algunas teorías de la enseñanza en los individuos, dada su importancia en el campo de la didáctica universitaria. Posteriormente, se abordan los temas de UML, software libre, su uso en la enseñanza, las estrategias didácticas y el ejemplo del CASE (Computer-aided software engineering) como forma de adquirir competencias en el área de la ingeniería de software.

4.1. MARCO HISTÓRICO

4.1.1. Antecedentes investigativos sobre estilos de enseñanza en la educación superior: El trabajo de Abello, Hederich y Hernández (2010) titulado *Caracterización de los estilos de enseñanza de una muestra de docentes de la Universidad Pedagógica Nacional*, es una investigación donde se elabora y valida un instrumento para identificar el estilo de enseñanza de los docentes universitarios a partir de las percepciones de los estudiantes. El instrumento se jerarquiza desde tres dimensiones que soportan el modelo establecido así: (i) social, (ii) control y gestión del aula y (iii) estrategias de aula, los cuales dan información relevante de los tópicos fijos y variables en el estilo que pueden presentar los docentes. De acuerdo con lo propuesto en la investigación le permite reconocer al docente su estilo de enseñanza, la forma en que los estudiantes lo ven, su propia visión del proceso de enseñanza para la toma de decisiones y mejoramiento de su modelo de enseñanza.

También se presenta la tesis doctoral de Camargo (2010) llamada *Dimensiones interactiva, discursiva y didáctica del estilo de enseñanza, el caso de las ciencias naturales*, la cual se refiere al estudio de la acción comunicativa de maestros de ciencias naturales del trabajo en el aula de clase, desde sus perfiles interactivo, discursivo y didáctico. Los resultados evidenciaron la identificación de siete tipologías de estilos de enseñanza desde un enfoque empírico y teórico. De igual forma, en la tesis se identifican correlaciones entre el estilo cognitivo de los profesores de la muestra y algunas de las tipologías formuladas.

Por otro lado, el trabajo de Rendón (2010) titulado *Los estilos de enseñanza en la Universidad de Antioquia*, es una investigación cuantitativa y descriptiva que reportan cómo se manifiestan los estilos de enseñanza en una muestra de profesores seleccionados de la Facultad de Educación de la Universidad de Antioquia. El resultado de la investigación demuestra un sentido correlacional, porque evalúa el grado de relación entre las respuestas de los docentes y las de sus estudiantes, y entre los estilos identificados y otras variables como el género de los profesores, la edad, los años de experiencia laboral, el nivel de formación de los profesores, el tipo de curso, el programa académico, el semestre en el que se desarrolla el curso. De acuerdo con el análisis se logran identificar los estilos de enseñanza de los profesores y los estilos de enseñanza en la educación superior, la categorización de sus indicadores y la construcción de instrumentos para evaluarlos.

De igual forma, la investigación de Gómez & Polanía (2008), cuyo título es *Estilos de enseñanza y modelos pedagógicos*, tiene como objetivo identificar los estilos de enseñanza contruidos por los docentes del área financiera en el programa de Ingeniería Financiera de la Universidad Piloto de Colombia y relacionarlos con los cinco modelos pedagógicos: (i) el tradicional, (ii) el conductista, (iii) el romántico, (iv) el cognitivo y (v) el crítico-social. Para la obtención de la información se diseñaron dos instrumentos de recolección de información: una guía de observación de las clases de cada profesor y una encuesta personal. La metodología es de tipo correlacional en este caso de un grupo de trece docentes del área financiera del programa de Ingeniería Financiera de la Universidad Piloto de Colombia.

Por su parte, Fonseca (2013) muestra un estudio titulado *Relación entre los estilos de enseñanza de la Maestría en Didáctica de las Ciencias y la Formación en Pregrado*, el cual surge colateralmente del interés del docente investigador en indagar sobre la relación existente entre los estilos de enseñanza y los de aprendizaje en torno al rendimiento académico de los estudiantes. Como producto de esta tesis doctoral, el autor creó un artículo científico que propone nuevas preguntas de investigación, una de éstas consiste en determinar la existencia de una relación entre la preferencia de los estilos de enseñanza y la formación de pregrado o formación profesional que presentan los docentes, específicamente los del Distrito Capital, en una de cuyas instituciones se realizó la investigación.

La tesis doctoral de Fonseca fue el producto de una investigación con metodología mixta, en un diseño exploratorio secuencial (Dexplos). En la parte cualitativa se realizó una investigación micro-etnográfica de las actividades de los docentes, específicamente el empleo de recursos y estrategias para construir el instrumento denominado test de estilos de enseñanza de Fonseca.

Finalmente, la tesis doctoral de Fonseca (2013) titulada *Efecto de los estilos de enseñanza y aprendizaje sobre el rendimiento académico*, es de tipo cuantitativo correlacional. Tiene como finalidad determinar la relación existente entre el estilo de enseñanza por preferencia de recursos y estrategias y la formación profesional inicial de docentes de la Secretaría de Educación del Distrito, en Bogotá.

De forma complementaria, también se pueden relacionar investigaciones específicas sobre el empleo de software libre en la enseñanza. El estudio de estas herramientas abarca numerosos estudios científicos, en este sentido se encuentra el realizado por Adell y Bernabé (2007), donde se describen los recursos de software libre, así como su filosofía y las ventajas de su empleabilidad. Estudios como el de Chavarría (2005), pone de manifiesto la importancia de este tipo de recursos como herramienta de apoyo al aprendizaje en los procesos formativos. Asimismo, el empleo del software libre fue propuesto como una herramienta docente en distintas ramas de la enseñanza universitaria, tal como lo observó Cuervo (2005) en su publicación titulada *La ingeniería de software libre aplicada a los proyectos informáticos*. En este sentido, estas investigaciones proponen el empleo de herramientas de software libre para la enseñanza de muchas áreas del conocimiento, así mismo describen las ventajas del empleo de este tipo de programas informáticos. De tal manera, se establece que el software libre confiere ventajas económicas, legales, científicas, formativas e incluso filosóficas.

En consecuencia, reportes académicos como el de Barreal & Ugolini (2014) titulado *El software libre como recurso didáctico en la enseñanza de métodos cuantitativos en los grados de economía y empresas*, confirman que “en la actualidad existen numerosos programas informáticos de código abierto o libres que pueden ser empleados como recurso docente en la enseñanza de la economía y de las ciencias empresariales” (Barreal & Ugolini, 2014, p. 21), claro está, sin que estos campos sean los únicos en los que el software libre se pueda aplicar.

De hecho, se advierte un importante cambio en la forma como se utiliza el software libre, dado que, al principio, las herramientas informáticas empleadas fueron principalmente de carácter privado. Sin embargo, con el paso de los años, éstas se fueron abriendo paso hacia modelos de software libre o código abierto. Estos programas informáticos se basan en múltiples formas de colaboración y estructuras organizativas que aumentan su competitividad a nivel académico y económico. Siendo este último factor muy importante para la distribución de gastos de las entidades educativas y del sector empresarial, poniendo de manifiesto que esta viabilidad económica añade un componente de análisis técnico y social de gran importancia.

Otro aporte que versa sobre el empleo de software libre es el de Hernández & Cuevas (2013) titulado Programas informáticos de uso libre y su aplicación en la enseñanza de la estadística. En esta publicación, los autores comentan que “fue aproximadamente a mediados del siglo XX cuando comenzaron a surgir iniciativas para eliminar las restricciones en el uso de programas de cómputo. Entre 1960 y 1970, no eran considerados un producto, sino un complemento indispensable para poder usarlas, por lo que era común observar que sus desarrolladores los compartían libremente con sus colegas” (Hernández & Cuevas, 2013, p. 168). En este sentido, se detectó un mercado potencial en la naciente industrial del software, el cual ha continuado evolucionando de manera paulatina hasta nuestros días. No obstante, los autores hacen una distinción importante, ya que es importante señalar que un programa de cómputo es de uso libre siempre y cuando el usuario tenga la libertad de, una vez obtenido, utilizarlo, copiarlo, estudiarlo, modificarlo y redistribuirlo libremente. Por lo tanto, no es correcto asociar los términos “programa de cómputo libre” con “programa de cómputo gratuito”, ya que, aunque conserve su carácter de libre, puede ser vendido comercialmente (Hernández & Cuevas, 2013).

Finalmente, Orjuela & Hurtado (2010) con su publicación Perfeccionamiento de un nuevo simulador interactivo, bajo software libre GNU/LINUX, como desarrollo de una nueva herramienta en la enseñanza y aprendizaje de la física, establecen que hay un sinnúmero de herramientas informáticas para el aprendizaje, en este caso de la física, sin embargo, “presentan ciertos tipos de restricciones de índole legal, las denominadas licencias; es decir, se necesita un permiso de manejo, limitando el papel educativo que deben tener éste tipo de herramientas didácticas” (Orjuela & Hurtado, 2010, p. 200). En consecuencia, es difícil predecir en qué dirección se moverá la informática como elemento fundamental en el avance de la ciencia, mientras la investigación en la enseñanza de la Física muestra que los avances tecnológicos no conllevan necesariamente a una mejora en el aprendizaje en estas áreas de conocimiento, pero a su vez se ha comprobado, que estas herramientas como el software libre son consistentemente necesarias para un adecuado aprendizaje de las ciencias exactas como lo puede ser la física en los estudiantes universitarios. Algunos tipos de licencias se ven en la Tabla 1.

Tabla 1. Licencias de software libre

LICENCIA	OBSERVACIONES
GPL	Permite Copyleft
AGPL	Obliga a distribución
BSD	Puede ser vendido
Apache	Exige reporte de los cambios
Creative commons	Se debe referenciar

Fuente. El autor

4.1.2. Teorías sobre la enseñanza en los individuos: Los seres humanos a través de la historia se han valido de diversas formas o estilos para dar solución a los diferentes problemas que surgen en su vida cotidiana. En las sociedades tradicionales, el saber era transmitido de forma oral a través de un proceso continuo y a través de una participación constante de las comunidades para mirar su propia cultura y cuyo objetivo ha sido mantener vivas las tradiciones por medio de símbolos o códigos culturales que resultaban ser una impronta; por ejemplo, como se refleja en los objetos elaborados a mano.

Siendo la educación una función tan natural y universal de la comunidad humana (Jaeger, 2000), la escolarización pertenece a un pequeño segmento de las formas o maneras en que una cultura la valida, según definición del concepto la cultura engloba la totalidad de las manifestaciones y formas de vida que caracterizan un pueblo. Cabe destacar que, las artes y los oficios se mantienen en la sociedad, a pesar de la carencia de maestros en dicha actividad cuyo sector se ha desestimulado con la desventaja de perder todo el saber que encierra dicha formación. Las investigaciones acerca de los estilos de enseñanza se han dividido en dos dimensiones: pedagógica y psicológica (Rojas, Díaz, Vergara, Alarcón, & Ortiz, 2016). La dimensión psicológica relaciona el estilo de enseñanza con una manifestación del estilo cognitivo, de aprendizaje y pensamiento del personal docente durante su ejercicio (Zhang, 2009). Desde la dimensión pedagógica, se considera la enseñanza como un proceso reflexivo y consciente, e incluso efectivo, puesto que se adecúa a los requerimientos de la cualificación (Zhang, 2009).

Los estilos de enseñanza pueden detectarse como un patrón específico de creencias, necesidades y comportamientos que los cuerpos docentes representan en las clases (Virvou & Mondridou, 2001). Para clasificar estos estilos, se proponen cuatro categorías: (i) *Doctor*, moldea comportamientos por medio de refuerzos; (ii) *Experto*, operacionaliza la adquisición del conocimiento; (iii) *Entrenador*, pone su énfasis en la aplicación del conocimiento; (iv) *Humanista*, propicia y gestiona el autoconocimiento.

De igual forma, Vigotsky (1978) plantea la importancia que representan los instrumentos culturales en la construcción del saber y del conocimiento, los cuales están mediados por las diferentes interacciones sociales en su vida cotidiana y las cuales afectan al individuo y a la colectividad, en ese sentido la educación se reconoce como una práctica cultural que cumple una función en la sociedad y el cual es mantener el patrimonio cultural. De ahí que, el contexto social tiene una marcada influencia en el aprendizaje que sirven de fuente para el desarrollo y estructuración de los procesos cognitivos. Así mismo, Feldman (1980), Feldman & Goldsmith (1986) y Gardner (1999) exponen que desde el ámbito cultural las

determinaciones biológicas y culturales, la mente humana construye significados a partir de un proceso de codificación y decodificación de la información que percibe desde su contexto cultural.

Aunque el individuo en su desarrollo cognitivo, como en el proceso de aprendizaje y de construcción de conocimiento, construye una carga de significación, mediados por unas características culturales propias de cada sociedad como simbolismo e identidad, la vida cotidiana del individuo se mueve en una dinámica de interpretaciones de la realidad que lo circunda y se manifiesta a través de instrumentos, que dan cuenta de toda la actividad mental, la cual se comunica a partir de sus tradiciones.

En toda comunidad la educación juega un papel fundamental y cumple una función social con características propias, que se constituyen para mantener vivo el patrimonio cultural y que el legado de tradiciones ancestrales se conserve y se imponga en el tiempo. La actividad de enseñar es a la postre una disciplina de instrucción, ciencia empírica desde sus inicios y como lo plantea Meyers (2000) la instrucción directa, pre-organizada y con unos objetivos establecidos que garanticen las condiciones del aprendizaje.

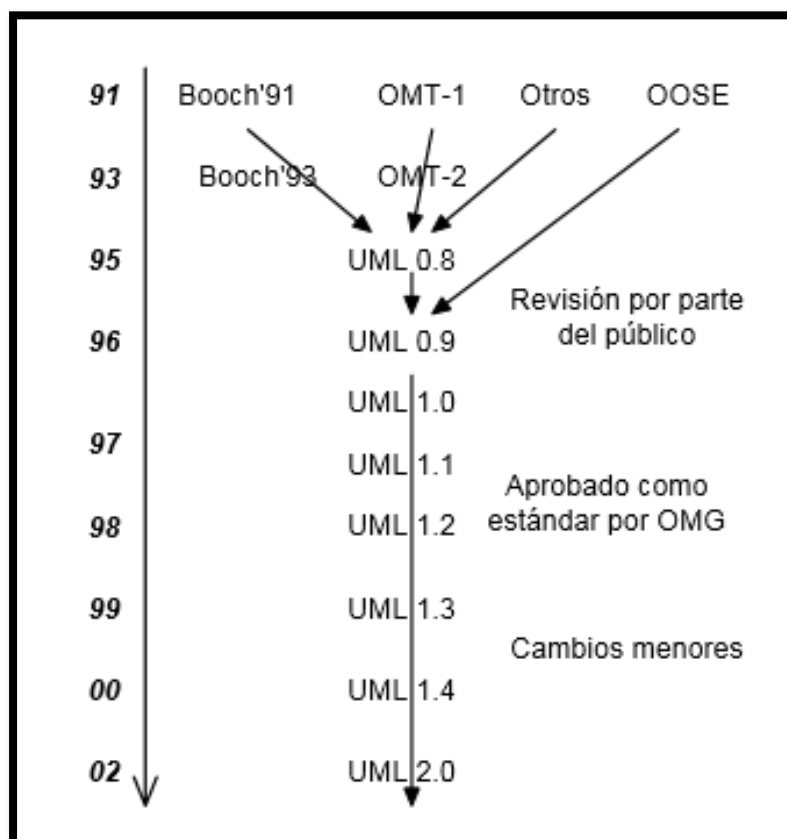
Para Gené (1998), el acto de enseñar se relaciona con la acción de mostrar a los estudiantes las formas de hacer, de pensar, de aprender, dentro de este concepto se incluyen los estilos de enseñanza que según diferentes autores Tabla 1 responden a las siguientes definiciones:

4.1.3. Antecedentes sobre UML: En el desarrollo de software se han presentado diversos hitos muy importantes que hicieron crecer y evolucionar de manera significativa cada una de las tareas involucradas con este. UML, Ingeniería de software y Software Libre han tenido diversos cambios y crecimiento a través del tiempo que le han permitido al software tener mucha madurez y sobre todo eficiencia desde su origen hoy.

El sistema UML nació en 1994 de la mano de Rumbaugh quien más adelante junto con Booch lo unificarían con OMT lo cual generó el primer borrador en 1995, luego se uniría Jacobson quien daría nuevas ideas al nuevo modelo que venía creciendo a pasos agigantados, a partir de esto nuevas empresas aportarían nuevas ideas dando origen a la primera versión en 1997. Durante los últimos años ha evolucionado hasta que en el año 2000 se generó la versión 1.4 que fue una de las mejores de la época incluso se usa todavía hoy en día, esta proporciona diversos tipos de diagramas que respaldan el análisis y diseño de software de forma muy

completa, en 2002 se desarrollaría la versión 2.0 la cual va creciendo y evolucionando hasta el presente (Hernández, 2012). La figura 1 muestra la evolución de UML.

Figura 1. Evolución de UML



Fuente: Hernández (2012)

UML 1.0 y 2.0. creado en 1991 UML 1.0 evolucionó hasta su máxima versión 1.5 en el año 2000 orientado a programación orientado a objetos, en este tiempo incorporó diagramas acordes a las necesidades de programación de la época y del recurso tecnológico disponible, a medida que fue creciendo y evolucionando la tecnología y nacieron las metodologías de desarrollo se necesitó tener un nuevo modelo que incorporara todos estos nuevos paradigmas, en respuesta a esto nació el modelo 2.0 en el año 2002 que además de soportar los diagramas de su versión padre tiene nuevos y mejorados elementos integrados con el diseño de software, versiones 2.1, 2.2 y 2.3 vieron su luz en 2007, en 2011 la versión 2.4 y para la actualidad la versión 2.5, estas mejoras se adaptaron a la evolución del desarrollo de software y las

nuevas tecnologías que requerían nuevos modelos para su creación, actualmente la versión 2.8 ha incorporado mejoras en cuanto a desarrollo móvil, web y en la nube (Difference Between, 2017).

4.1.4. Antecedentes sobre el Software Libre y su uso en la enseñanza: El concepto sobre el Software Libre nació en los años 50 en el MIT, por los años 70 las comunidades de desarrolladores e ingenieros informáticos compartían el software que creaban entre ellos, en esa misma época Richard Stallman llega al MIT donde fundaría la FSF para reforzar la política libre en el software, para el año de 1975 se desarrollaría el primer proyecto de sistema operativo libre para una maquina llamada Altair 8800, en 1984 se lanzaría el proyecto GNU llamado así por la definición “GNU no es Unix” (Rueda, 2004).

El 1991 Linux Torvalds desarrollaría la primera versión de su sistema operativo Linux el cual mejoraría significativamente los meses siguientes incorporando muchas mejoras, en 1992 se incorporan paginación de disco y mejoras de memoria, para 1992 Linus se acoge al modelo de licencia GPL. En 1994 y 1995 incorporaría mejoras de calidad a su sistema y compatibilidad entre el hardware existente de la época siendo una versión muy estable y funcional contra sus competidores (Rueda, 2004), es aquí donde se genera el punto de partida de la adopción de software libre entre usuarios y las empresas generando una nueva concepción de que es y cómo usar el software. Desde 1995 hasta la fecha los diferentes desarrolladores de todo el mundo y empresas dedicadas a crear aplicaciones han generado diversas versiones de sistemas operativos y software que se usa en muchos campos de acción y diversas ramas del saber evolucionando a cada día.

En la actualidad ya son muchas las organizaciones que producen y promueven el software libre no sólo para cualquier tipo de personas, sino para utilidades en el sector educativo. Es el caso de la organización GNU Operating System (GNU Operating System, 2017) que aboga por defender la libertad de escogencia de los métodos y formas de enseñanza. Para esta organización, *“All these programs are released under a license that is granted for zero price and does not expire. While we are glad that schools can also save money with these programs, avoiding miseducation (teaching dependence on nonfree software) is a more important imperative: when we say these programs are free software, we are talking about freedom, not price. It means that you are free to use them constructively, either alone or in a community, while respecting the freedom of others”* (GNU Operating System, 2017).

La relevancia acerca de la utilización del software libre no es poca. La escuela y las universidades son el lugar donde se están educando a las próximas generaciones los valores de nuestras sociedades, pero también el conocimiento que necesitan para encontrar un lugar en esa sociedad. También es allí donde aprenden cómo usar la computadora, cómo interactuar con Internet, las redes, las plataformas, entre muchos otros ambientes. Las escuelas y universidades deberían enseñar a los individuos a ser buenos miembros de la comunidad y los establecimientos educativos tendrían que tener entre sus funciones las capacitaciones sobre productos que son libres y también sobre los que no lo son.

Durante demasiado tiempo, la utilización de software se ha visto dentro de los sistemas de educación en muchos países, donde Colombia no es la excepción, no obstante, se ha reducido a la mera capacitación del usuario. Por supuesto, no todos los estudiantes aspiran a una carrera en informática, sin embargo, es esencial que cada uno comprenda los mecanismos y lo que está en juego, para poder reclamar el mundo digital para él o ella en lugar de estar sujeto a ella. La tecnología debe enseñarse como una materia escolar por derecho propio, proporcionando a los estudiantes la capacidad de comprender y adaptarse a las diferentes situaciones que encontrarán en su vida personal y profesional. El software libre es el único que permite completamente esto, precisamente porque está abierto y puede estudiarse, modificarse y compartirse.

El software libre y los formatos abiertos tienen un papel completo que desempeñar en la educación, ya que no se supone que este último favorezca a este o aquel proveedor de soluciones patentadas. Bajo la cobertura de preparar a los estudiantes para los productos de software que encontrarán en su vida profesional, las prácticas actuales refuerzan los monopolios y persisten en transmitir recetas ya preparadas en lugar de enseñar a ser autosuficientes. El uso de formatos abiertos y el licenciamiento de recursos educativos bajo una licencia libre permiten compartir y mejorar esos mismos recursos, contribuyendo así al desarrollo de una educación más ética. El software libre puede distribuirse sin restricciones, lo que permite a los docentes encontrar más recursos y crear sus propios materiales educativos y como, resultado de un esfuerzo colaborativo ejemplar de maestros de matemáticas y publicado bajo una licencia gratuita.

4.1.5. Antecedentes de la ingeniería de software: Tiene sus inicios en los años 50 donde incorpora el método científico y los desarrolladores de software se basaban más en su experiencia y conocimientos propios sin documentar ni generar escalamiento en el software, en los años 60 se incorporan nuevos conceptos como la interacción hombre-máquina, multiprogramación, multiusuarios y creación de gestores de bases de datos, es un punto importante ya que el software cambia su imagen de ser un simple elemento a un producto que debía ser tenido en cuenta para el crecimiento de las empresas, aunque todavía debía corregirse mucho y los costos de un proyecto de software eran incalculables y variables, para tratar de mitigar esto en 1968 se generó la primera conferencia de software donde se trató la "crisis del software" como tema principal y se generaría por primera vez el termino ingeniería de software. Para los años 70 nuevos desarrollos en redes de computadores, sistemas distribuidos y demanda de acceso a los datos hizo que se incorporaran diversas fases en la creación de software llamándola programación estructurada, surgen modelos nuevos como la tradicional "cascada" (Frasser, 2004).

En los años 80 ya se encontraban conformados equipos de desarrollo y la industria del software ya tenía forma, incluso se hablaba de una cuarta generación, nace la programación orientada a objetos y nuevos lenguajes de programación más fáciles de utilizar y codificar, en los años 90 para el mejoramiento de la programación orientada a objetos se crea el lenguaje de modelado UML y se generan procesos comerciales enfocados a este paradigma llamada RUP, se crean nuevos modelos de desarrollo como el espiral, análisis de riesgo, iterativo e incrementales (Frasser, 2004) (Zaragaza & Nogueras, 2008).

En la actualidad con la incorporación de nuevos dispositivos móviles, web y tecnologías emergentes como la nube hacen que el software incorpore nuevas formas de creación, además temas como la seguridad, usabilidad y confiabilidad son más exigentes y solicitadas, la conectividad también influye en el crecimiento del software y la incorporación de negocios digitales hacen que cada día mejore en cuanto a desarrollo y procesos de desarrollo (Zaragaza & Nogueras, 2008).

4.2. MARCO CONCEPTUAL

4.2.1. Estrategias didácticas para la educación superior: Las estrategias de enseñanza de la escuela secundaria suponen que se está trabajando con y enseñando a adultos jóvenes. Los ejemplos acerca de la enseñanza en diferentes niveles dentro de un distrito escolar indican que se debe ser flexible y cambiar la forma en que se presenta el material en función del auditorio. Lo que funciona con

los estudiantes de primaria no siempre va con los niños de secundaria, lo mismo para jóvenes en la universidad.

Siguiendo a Dewey (1993) con relación a la manera como las personas piensan y su forma de aprender, se propone una pregunta base fundamental: ¿Qué es pensar? Para Dewey, el pensamiento es algo innato en las personas que no se enseña, simplemente viene con nosotros. Incluso, si alguien desea detener el pensamiento no podrá hacerlo, porque estará pensando en dejar de pensar, suena algo repetitivo, pero el pensamiento humano siempre está presente incluso cuando dormimos, es la corriente de la conciencia lo que define esta manera incesante de maquinar o de traer a la mente las cosas que suceden (Dewey, 1993).

Para dar un sentido, un orden a los pensamientos, Dewey plantea la idea de un pensamiento reflexivo, el cual es una cadena de pensamientos que van uno tras otro hasta llegar a una conclusión deseada. “La reflexión no implica tan solo una secuencia de ideas, sino una consecuencia, esto es, una ordenación consecucional en la que cada una de ellas determina la siguiente como su resultado. (...) En lenguaje técnico, un término del pensamiento. Cada término deja un residuo que es utilizado en el término siguiente” (Dewey, 1993, pág. 22). Lo anterior quiere decir, que el pensamiento reflexivo debe apuntar a una conclusión, es por ello por lo que no es gratuita las expresiones “piénsalo bien” o “piénsalo despacio”; se quiere decir que aquella cadena de pensamientos debe llegar a un punto que es a donde se quiere llegar, a la respuesta de las preguntas que nos asaltan.

Y es por esto mismo que para Dewey el pensamiento reflexivo es el que nos encamina al pensamiento que impulsa la investigación y el sentido de la educación superior al cultivar un terreno para la creación de conocimientos. “Lo que constituye el pensamiento reflexivo es el examen activo, persistente y cuidadoso de toda creencia o supuesta forma de conocimiento a la luz de los fundamentos que lo sostienen y las conclusiones a las que tiende” (Dewey, 1993, pág. 25).

Ahora, ¿por qué el pensamiento reflexivo se constituye en un objetivo de las estrategias didácticas en la universidad? Para nadie es un secreto que aquello que nos diferencia de los animales es la capacidad de razonar y de reflexionar. En este punto, es valioso reconocer que existen al menos tres valores en el pensamiento humano: (i) nos libera de los impulsos e instintos naturales para que actuemos de forma deliberada, (ii) permite la creación de signos artificiales y la predicción de resultados, y (iii) permite cambiar significados y significantes a los signos creados. “Estos tres valores, con su efecto acumulativo, constituyen la diferencia entre una vida verdaderamente humana y la existencia tal como la viven los animales, limitados por la sensación y el apetito” (Dewey, 1993, pág. 36).

No sabemos si por desgracia o por fortuna, este proceso descrito anteriormente es homogéneo para todos. Sí, existen unas generalidades, pero como hemos visto hasta ahora, las diferenciaciones estarán presente en todos y cada uno de los aspectos de nuestras vidas, luego, ¿Será que todos tenemos las mismas capacidades para aprender cosas nuevas? A priori podemos inferir que no. De allí a que posiblemente surjan diferentes enfoques pedagógicos o educativos para encontrar las mejores formas de enseñanza.

4.2.2. ¿Cómo enseñar la ingeniería de software?: La ingeniería de software es un componente central de muchos programas de ingeniería informática. En los cursos de ingeniería de software, se les enseña a los estudiantes a aplicar sus habilidades de programación y desarrollo para resolver un problema de mayor escala. La resolución de este problema implica el desarrollo de una comprensión del problema desde la perspectiva del cliente, así como un análisis de alternativas de solución.

Desafortunadamente, en muchos casos, el curso de ingeniería de software se ofrece tarde en el plan de estudios y generalmente en los niveles superiores. Esto hace que sea difícil para los estudiantes aplicar el conocimiento que han aprendido efectivamente en el tope y otros proyectos académicos. Los estudiantes a menudo comentan que habría sido "bueno saber esto" antes de tomar decisiones equivocadas en sus proyectos culminantes. Por lo tanto, para tener éxito, los componentes de la ingeniería de software deben enseñarse antes en el plan de estudios de pregrado. Sin embargo, este cambio a un nivel anterior plantea problemas pedagógicos.

Un curso de ingeniería de software es diferente de otros cursos, dado que nuestros otros cursos tratan sobre hechos tangibles: la semántica de una instrucción de lenguaje de programación, la complejidad de un algoritmo o las reglas de un negocio; pero la ingeniería de software trata con ideas intangibles: qué metodología se puede usar en un determinado dominio de aplicación, cómo se puede tratar con sistemas heredados, qué heurística se puede aplicar para diseñar un conjunto de pruebas efectivo. La falta de claridad general del contenido del curso de ingeniería de software pone nerviosos a estudiantes y profesores.

Es así como en este contexto nace y se hace presente el software CASE (Computer-aided software engineering) o ingeniería de software asistida por computadora, que ayuda a aliviar los procesos pedagógicos tanto para estudiantes como para profesores. CASE ofrece un entorno tangible y práctico en el que los estudiantes pueden probar las ideas discutidas en otras partes del curso. Pueden ver estas ideas en acción mientras el software comprueba su trabajo en busca de consistencia e integridad.

Hasta hace poco, las herramientas CASE eran difíciles de encontrar. Algunos libros de texto de ingeniería de software proporcionaron un conjunto limitado de herramientas. En muchos casos, estas herramientas fueron escritas por el autor del libro de texto como una idea de último momento, y tenían poca o ninguna versatilidad en el mundo real. (Una excepción notable es el libro *Software Engineering With C ++ and CASE Tools* de Michael Pont, publicado por Addison-Wesley (Pont, 2015). Para Rodríguez (2005), “estas son tecnologías para automatizar el desarrollo y mantenimiento del software, combinan herramientas de software y metodologías, y deben constituir un conjunto integrado que automatice todas las partes del ciclo de vida y ahorren trabajo a los especialistas” (Rodríguez, 2005, p. 32).

Para finalizar, se quiere advertir que es así como se va constituyendo el tema de investigación de este proyecto, en donde se busca establecer la pertinencia del uso de software libre en estas áreas que suponen aún muchas preguntas por resolver en materia de enseñanza y aprendizaje.

4.3. MARCO LEGAL

4.3.1. Licencias de software libre. Existen diversos tipos de licenciamiento de software libre que se enfocan en diversas características propias del uso, modificación y distribución del sistema, ya sea desde el uso personal o comunitario y los fines con o sin ánimo de lucro, las más importantes según (Universidad de castilla, 2015):

- GNU – GPL. Esta licencia facilita el usar, estudiar, compartir y actualizar el producto software, la idea principal de esta licencia es protegerlo de que en algún momento se vuelva privativo, lo anterior se garantiza estableciendo que quien utiliza o realiza cambios en el software deberá compartir el código fuente como el binario generado.
- AGPL. Es una licencia GPL pero que tiene la característica que es usada para software que ofrece servicios por redes de computadores o en línea, más que todo para servicios web.
- BSD. Usado en sistemas operativo BSD es básicamente una renuncia a la garantía del producto para dar reconocimiento a la autoría de las mejoras realizadas sean buenas o malas, se permite la redistribución y son mezcladas habitualmente con la licencia GPL.

- MPL. Usada por Mozilla Firefox se creó con el fin de mitigar un poco las facilidades de la licencia GPL siendo menos permisiva que BSD, actualiza muchos puntos de GPL beneficiando más al desarrollador y al nuevo producto generado conservando la libertad de distribución.

4.3.2. Ley 23 de 1982 de derechos de autor. La ley tiene en cuenta: “obras científicas, literarias y artísticas las cuales comprenden todas las creación es del espíritu en el campo científico, literario y artístico, cualquiera que sea el modo o forma de expresión y cualquiera que sea su destinación”, con respecto a la afirmación anterior se tiene en cuenta que el registro de una propiedad intelectual se hace en la Dirección Nacional de Derechos de Autor del Ministerio del Interior, se anexa una copia y se llena el formulario requerido para el registro el cual es gratis, como el software se compara a la obra literaria también se debe registrar ante la Dirección Nacional de Derechos de Autor.

Sin duda, el uso de Software Libre implica una serie de concepciones que son difíciles de entender al inicio, ya que en principio las personas se enfrentan a un mundo legal que está delimitado por las licencias, que en muchos de los casos interponen el empleo de recursos económicos antes que el empleo de las herramientas. Por lo tanto, la normatividad empieza a obrar en unas esferas que hasta ahora, e incluso todavía, son desconocidas por muchas personas.

No obstante, existe un grupo de personas cada vez mayor que está volcando hacia las nuevas alternativas diferentes a las licencias. El Ministerio de Educación (MEN, 2018) ha puesto en la mira este importante cambio de mentalidad, y registra que “unos por convicción política y otros por beneficios económicos, han decidido no pagar por las licencias de programas para computador, ni usarlos piratas” (MEN, 2018); esto como respuesta de quienes han migrado al uso de programas de software libre, como una alternativa que se ofrece para que puedan ser modificados por cualquiera y la mayoría se pueda descargar sin costo desde internet.

Aún falta divulgar efectos legales e impactos del software libre en Colombia. Al respecto, el Ministerio de Educación (MEN, 2009) advierte como la difusión de los beneficios de trabajar con software libre no ha sido masiva, la comunidad trabaja en varios frentes para ampliar el mercado de usuarios. “Uno de ellos son las JSL (jornadas de software libre) que se hacen en diferentes ciudades de Colombia, en las que se trabaja desde el ámbito político, velando porque desde el Estado no se restrinja el espacio para los desarrolladores de software libre” (MEN, 2009). En este sentido, se aboga para que el Gobierno no se vaya a ir por un modelo restrictivo como el de establecer que todas las entidades del Gobierno deban usar el sistema

operativo Microsoft, por lo que el impacto de estas jornadas busca que existan actores en la política que abran espacios, sobre todo en lo jurídico.

Como respuesta a estos mecanismos de presión en el ámbito jurídico, el Acuerdo 331 de 2006 del Concejo de Bogotá, le propone al Alcalde Mayor el expedir una “Política de Promoción y Uso del Software Libre en el Sector Central, el Sector Descentralizado y el Sector de las Localidades del Distrito Capital y se trazan lineamientos sobre el contenido de dicha política” (Concejo de Bogotá, 2006). En este acuerdo, se pone de manifiesto que “una de las herramientas con mayor potencial para estos efectos es el software libre, por sus implicaciones en la democratización de la información, la promoción de la autonomía tecnológica, el acceso al conocimiento, el impulso a la industria y mano de obra locales, la transparencia y la modernización de la gestión pública, la reducción de costos por uso de licencias y la seguridad informática” (Concejo de Bogotá, 2006).

El Concejo de Bogotá y el Ministerio de Educación Nacional, están de acuerdo en que la utilización del código abierto permite adelantar una amplia gama de tareas, desde las más complejas, como desarrollo de aplicaciones, producciones de cine y televisión, servidores de internet y operaciones de misión crítica, hasta las más sencillas, como aplicaciones de ofimática, entretenimiento y usos en el hogar y la oficina. De esta forma, al revisar los informes sobre software libre que elaboran empresas, gobiernos y organizaciones internacionales, se observa que los expertos consideran, por ejemplo, el Linux, como un sistema operativo muy estable, que no presenta problemas para interactuar con otros sistemas, gracias a la arquitectura abierta y a la flexibilidad.

En consecuencia, se promueve la lógica de la cooperación y la necesidad de compartir el conocimiento para reducir la inequidad en el avance tecnológico, por lo cual los programadores autorizan el uso de sus programas sin reservas, tal como lo hizo Stallman (2016), por ejemplo, cuando creó y autorizó el uso de la licencia GPL. Además, los autores originales nunca quedan relegados por otros o por una empresa más grande que use sus productos, puesto que se convierten en los ejes del proyecto al ser quienes recopilan los reportes de errores, los requerimientos de los usuarios y las contribuciones de los demás programadores.

5. CARACTERIZACION DE LA POBLACION

La población objeto del siguiente trabajo está enfocada en estudiantes de educación superior que se encuentran cursando asignaturas relacionadas con ingeniería de software y/o programación. La clasificación viene dada de la siguiente manera:

- Institución: Fundación universitaria Unipanamericana
- Semestres: Cuarto y Quinto
- Carrera: Ingeniería de sistemas, software y telecomunicaciones
- Cursos: Algoritmos y programación, gestión y calidad de la información, diseño de investigación.
- Cantidad de estudiantes: 88 estudiantes universitarios de edades de 16 años a 32 años.

6. PROCEDER METODOLOGICO

6.1. ENFOQUE DEL EJERCICIO INVESTIGATIVO

El estudio de la investigación se centrará en evaluar si es viable la inclusión de software libre en la enseñanza de UML en ingeniería de software, esta investigación es de carácter cualitativa e interpretativa, se basa en el significado que los estudiantes de los cursos definidos de la fundación universitaria panamericana otorgan al caso de estudio.

6.2. METODOLOGÍA CENTRAL Y EMERGENTE

La metodología que se utilizará será:

- Nivel: Perceptual
- Objetivo: Exploratorio
- Tipo de investigación: Cualitativa e interpretativa

6.3. DISEÑO DE INSTRUMENTOS

Se utiliza la encuesta como el medio más rápido y claro para recolectar la información de las opiniones de los estudiantes, este elemento de recolección de información permite centrar el objetivo de la investigación.

6.3.1. Categorías: Las categorías para validar la información son a nivel de:

- Acuerdo o desacuerdo
- Frecuencia de utilización
- Importancia de su aplicación
- Valoración del producto
- Probabilidad de uso

6.3.2. Variables: Las variables clasifican las preguntas del trabajo en:

- Edad
- Sexo
- Nivel de satisfacción del estudiante
- Conocimientos de aplicativos de software libre
- Uso de aplicativos de software libre en general
- Nivel de satisfacción del uso de software libre para aprender UML

6.4. TRATAMIENTO DE LA INFORMACIÓN

6.4.1. Sistematización: Los datos serán almacenados en un archivo plano en hoja de calculo

6.4.2. Codificación: Serán clasificados según las variables utilizadas en tablas de resultado según la muestra poblacional.

6.4.3. Análisis: El análisis de la información se realiza según la escala de Likert, a parte del mecanismo, en este caso, al ser cualitativo, cruzando las variables con las categorías y obtener los resultados para el cumplimiento de los objetivos.

6.4.4. Presentación de los datos: Los datos serán presentados en tablas y graficas de resultados en pastel

6.5. TRABAJO DE CAMPO

Se realizará un cuestionario a 88 estudiantes de 3 cursos en la fundación universitaria panamericana para analizar la información y obtener las conclusiones de la investigación.

6.6. PILOTAJE

Se considera la utilización de un cuestionario para evaluar los niveles de inclusión de aplicativos de software libre para ingeniería de software, es decir, estimar que tan válidos y efectivos serán los resultados obtenidos.

En este sentido, se utilizaron 6 variables para validar diferentes aspectos fundamentales a la hora de enseñar ingeniería de software.

7. DESARROLLO DE LA PROPUESTA

7.1. ESTRUCTURA

Se realizó un cuestionario de 30 preguntas donde se busca conocer la inclusión de software libre en aprendizaje UML, lo anterior en dirección a analizar la confiabilidad, nivel de eficacia, eficiencia y satisfacción que produce en el usuario, su uso y aprendizaje estimado.

Las preguntas se clasifican según las variables de la siguiente manera:

7.1.1. Variable Edad: Pregunta edad según rangos determinados.

7.1.2. Variable Genero: Pregunta genero según opción Masculino o Femenino

7.1.3. Variable nivel de satisfacción del estudiante:

- P9 - ¿Considera su formación adecuada con el uso de software libre?
- P18 - ¿Está de acuerdo con las políticas de su institución educativo en cuanto al uso de software libre?
- P19 - Desde su punto de vista, ¿El nivel de equipos de cómputo de la institución permitiría la inclusión de software libre?
- P20 - ¿El nivel de apropiación de sus docentes en la enseñanza de software libre es el adecuado?
- P25 - ¿Se ha sentido a gusto usando software libre?
- P26 - ¿Que tanto recomendaría el uso de software libre en su aprendizaje?

7.1.4. Variable conocimientos de aplicativos de software libre:

- P1 - ¿Entiende el concepto de ingeniería de software?
- P2 - ¿Entiendo el concepto de software libre?
- P7 - Señale algunas ventajas que conozca del software libre UML
- P11 - Señale si conoce alguna de estas herramientas
- P15 - Del siguiente listado, ¿Considera interesante del software libre para su utilización?
- P22 - ¿Siente que requiere más conocimiento del uso de software libre?

7.1.5. Variable uso de aplicativos de software libre en general:

- P3 - ¿Usa software libre frecuentemente?
- P5 - ¿Utiliza algún software en el diseño de UML?
- P14 - ¿En algún momento ha utilizado software libre en el diseño de diagramas UML?
- P16 - ¿Preferiría utilizar aplicativos de software libre en su aprendizaje que software propietario?
- P17 - ¿Estaría de acuerdo en que su empleo se utilizara únicamente software libre?
- P21 - ¿Encuentra útil el uso de software libre en la enseñanza de ingeniería de software?
- P23 - ¿Ha utilizado algún software libre para la recolección de requisitos en la fase de análisis en ingeniería de software?
- P24 - ¿Ha utilizado algún software libre para realizar diagrama de clase en la fase de diseño en ingeniería de software?
- P29 - ¿Que tan dispuesto estaría en usar otro sistema operativo que no sea Microsoft Windows si de eso dependiera el uso de software libre en su proceso de enseñanza de ingeniería de software?

7.1.6. Variable nivel de satisfacción del uso de software libre para aprender UML

- P4 - ¿En la asignatura de ingeniería de software o programación entendí la aplicabilidad de UML?
- P6 - ¿Está de acuerdo con el uso de software libre en la enseñanza?
- P8 - ¿Considera importante y útil se utilice software libre en enseñanza de UML?
- P10 - ¿Sugeriría a su institución educativa implementar la enseñanza de ingeniería de software con la utilización de software libre?
- P12 - ¿Considera importante la inclusión de software libre en la enseñanza de ingeniería de software?
- P13 - ¿Su nivel de formación se vería afectado si se enseña con software libre?
- P27 - ¿Cual es nivel de confianza para su proceso aprendizaje si únicamente se utiliza software libre en el aprendizaje de UML?
- P28 - ¿Cuál es su opinión respecto a si debería mezclarse la enseñanza de ingeniería de software usando software libre y software privativo?
- P30 - ¿Se sentiría satisfecho con el hecho que el software libre en algunas ocasiones tiene menos herramientas que un software privativo pero que de igual manera contiene lo necesario para el diseño de diagramas UML en fase de diseño?

7.2. RESULTADOS

7.2.1. Evaluación de los resultados: se analizan los resultados encontrados en las encuestas realizadas a los estudiantes y se evalúan según las variables determinadas en el proceder metodológico, con el fin de determinar puntos importantes en cada pregunta.

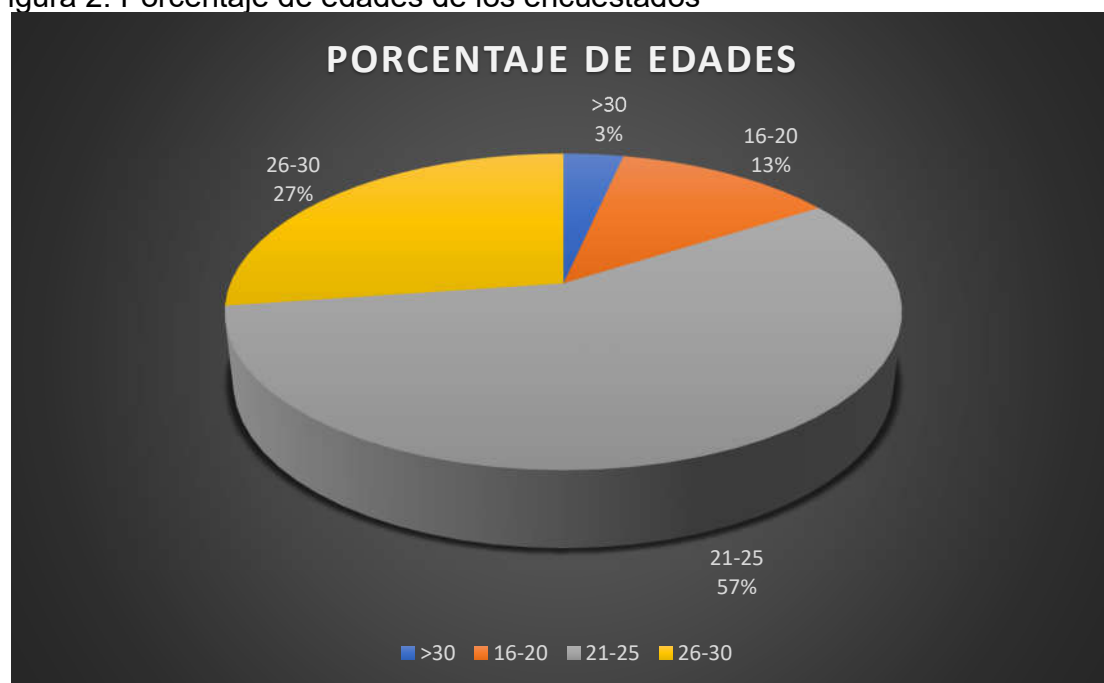
7.2.1.1. Edad: Se identifica que el (57%) de los estudiantes encuestados tienen edades entre 21 años y 25 años como se muestra en la Tabla 2 y representado en la Figura 2.

Tabla 2. Edades de los encuestados

EDAD	CANTIDAD	PORCENTAJE
>30	3	3
16-20	11	13
21-25	50	57
26-30	24	27

Fuente. El autor

Figura 2. Porcentaje de edades de los encuestados



Fuente. El autor

7.2.1.2. Sexo: Se identifica que el (86%) de los estudiantes encuestados son hombres contra un (14%) de mujeres, los resultados se muestran en la Tabla 3 y Figura 3.

Tabla 3. Cantidad de genero de los encuestados

GENERO	CANTIDAD	PORCENTAJE
F	12	14
M	76	86

Fuente. El autor

Figura 3. Porcentaje de genero



Fuente. El autor

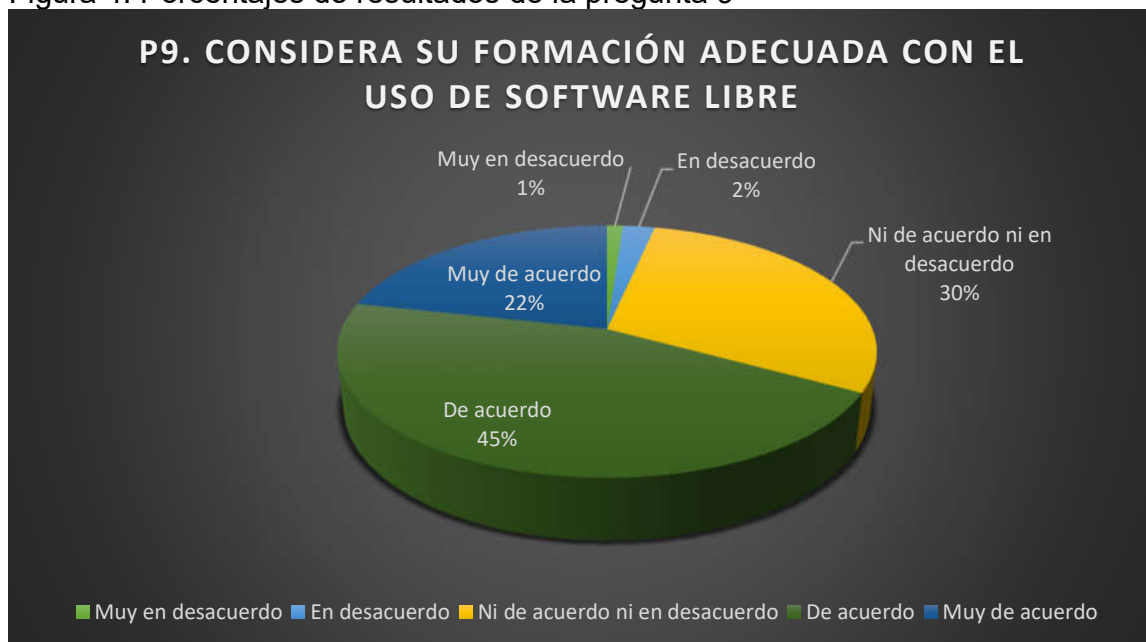
7.2.1.3. Nivel de satisfacción del estudiante: Los estudiantes encuentran en un (45%) estar de acuerdo en cuanto a si su formación académica fue adecuada usando software libre, los resultados se muestran en la Tabla 4 y en la Figura 4, por otro lado el (52%) de los estudiantes considera no estar de acuerdo ni en desacuerdo con las políticas de la institución educativa en cuanto al uso de software libre, este resultado evidencia un desconocimiento de las normas y procedimientos en el uso de este tipo de software, los resultados se muestran en la Tabla 5 y representados en la Figura 5.

Tabla 4. Resultados pregunta 9

P9. Considera su formación adecuada con el uso de software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	2	2
Ni de acuerdo ni en desacuerdo	26	30
De acuerdo	40	45
Muy de acuerdo	19	22

Fuente. El autor

Figura 4. Porcentajes de resultados de la pregunta 9



Fuente. El autor

Tabla 5. Resultados pregunta 18

P18. ¿Está de acuerdo con las políticas de su institución educativo en cuanto al uso de software libre?	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	7	8
Ni de acuerdo ni en desacuerdo	46	52
De acuerdo	30	34
Muy de acuerdo	4	5

Fuente. El autor

Figura 5. Porcentajes de resultados de la pregunta 18



Fuente. El autor

Desde otra perspectiva referente a la satisfacción del estudiante, se encuentra que el (35%) de los encuestados afirma estar de acuerdo con que los niveles de recursos tecnológicos de los equipos de cómputo de la institución son adecuados para la inclusión de software libre, un porcentaje algo bajo para la importancia de las necesidades de instalación del producto, los resultados se muestran en la Tabla 6 y la Figura 6. Por otra parte, analizando el nivel de conocimientos de los docentes referente a la apropiación y enseñanza de software libre se encuentra que el (39%) de los estudiantes está de acuerdo con que es adecuado, los resultados se evidencian en la Tabla 7 y Figura 7.

Tabla 6. Resultados pregunta 19

P19. Desde su punto de vista, el nivel de equipos de cómputo de la institución permitiría la inclusión de software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	2	2
En desacuerdo	12	14
Ni de acuerdo ni en desacuerdo	22	25
De acuerdo	31	35
Muy de acuerdo	21	24

Fuente. El autor

Figura 6. Porcentajes de resultados de la pregunta 19



Fuente. El autor

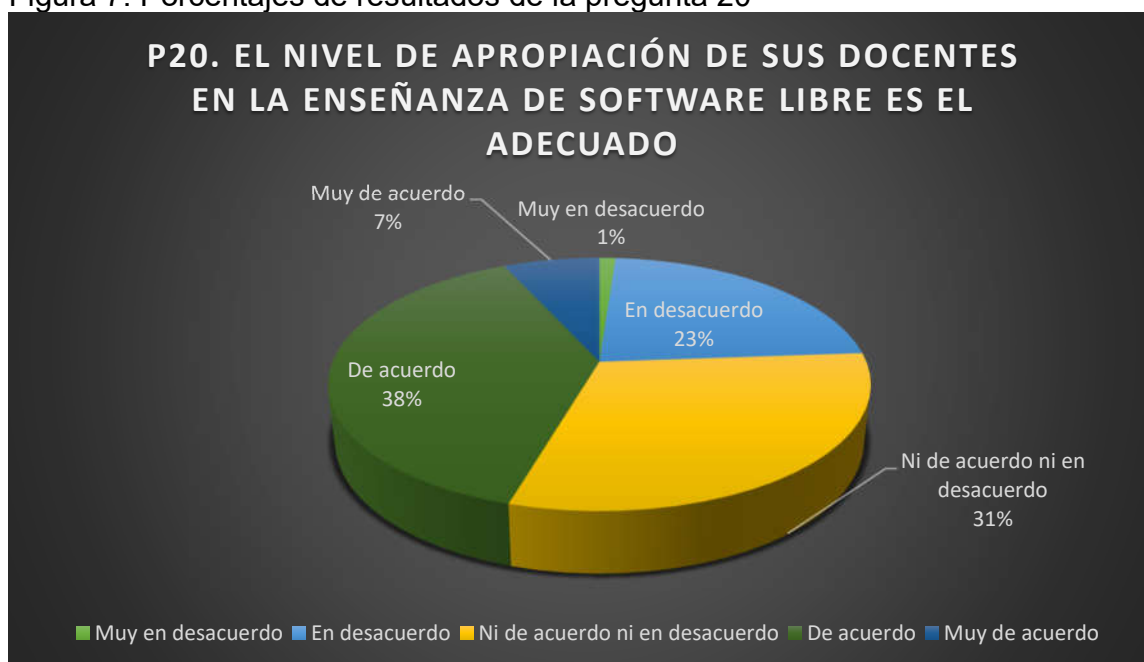
Tabla 7. Resultados pregunta 20

P20. El nivel de apropiación de sus docentes en la enseñanza de software libre es el adecuado	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	20	23
Ni de acuerdo ni en desacuerdo	27	31
De acuerdo	34	39
Muy de acuerdo	6	7

Fuente. El autor

A niveles de gustos y recomendaciones, se encontró que el (43%) de los encuestados está de acuerdo en que se sienten a gusto usando software libre y el (57%) recomendarían su uso en el aprendizaje, los resultados se evidencian en las Tabla 8, Tabla 9 y las Figura 8, Figura 9.

Figura 7. Porcentajes de resultados de la pregunta 20



Fuente. El autor

Tabla 8. Resultados pregunta 25

P25. Se ha sentido a gusto usando software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	2	2
En desacuerdo	2	2
Ni de acuerdo ni en desacuerdo	15	17
De acuerdo	38	43
Muy de acuerdo	31	35

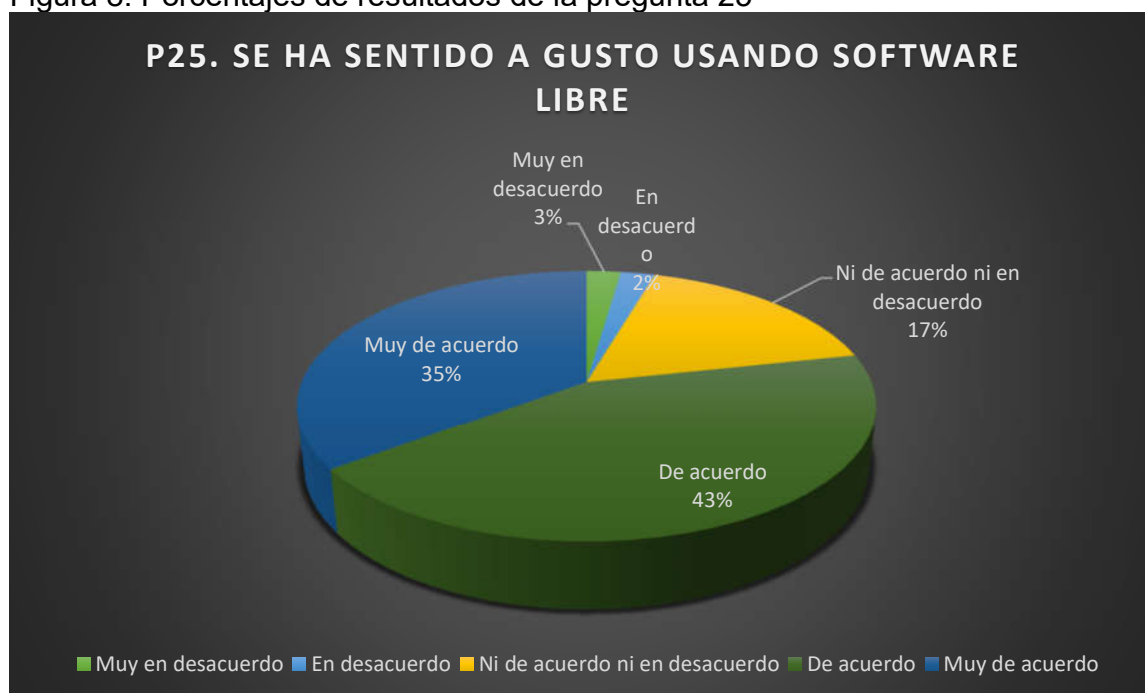
Fuente. El autor

Tabla 9. Resultados pregunta 26

P26. Que tanto recomendaría el uso de software libre en su aprendizaje	CANTIDAD	PORCENTAJE
Muy en desacuerdo	0	0
En desacuerdo	2	2
Ni de acuerdo ni en desacuerdo	11	13
De acuerdo	50	57
Muy de acuerdo	25	28

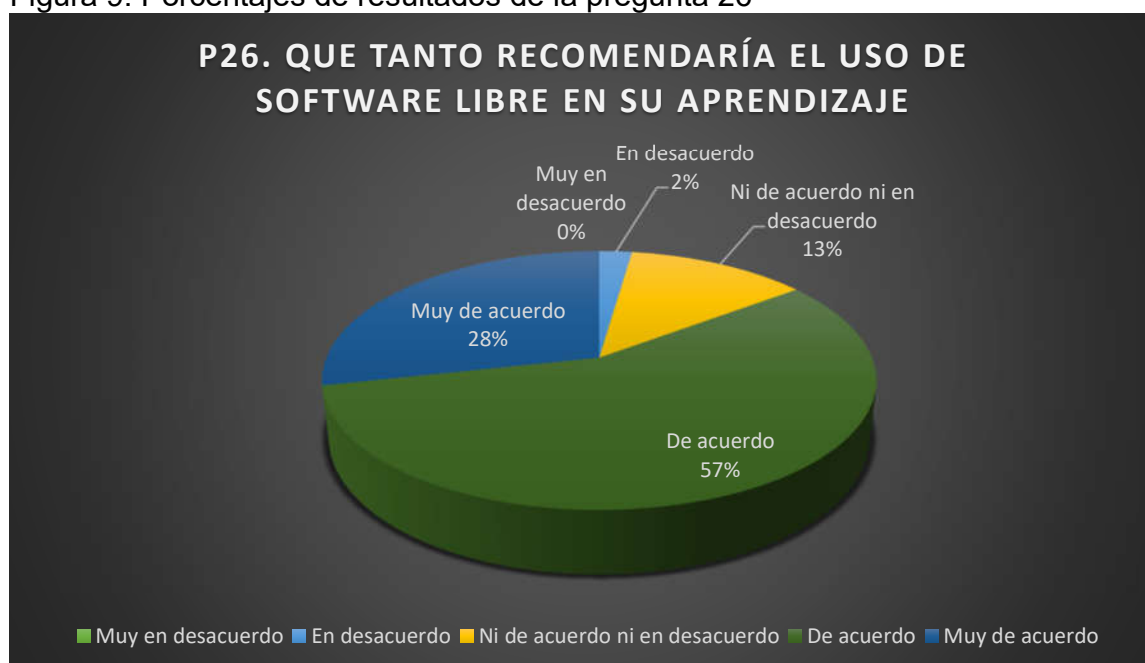
Fuente. El autor

Figura 8. Porcentajes de resultados de la pregunta 25



Fuente. El autor

Figura 9. Porcentajes de resultados de la pregunta 26



Fuente. El autor

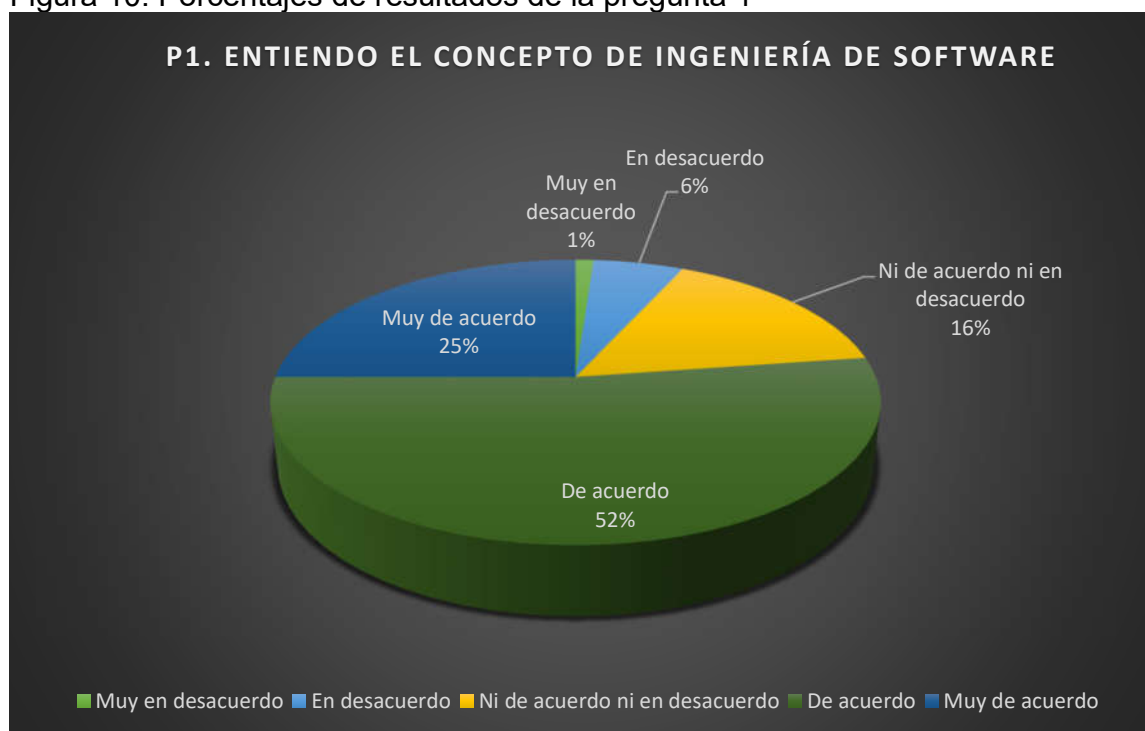
7.2.1.4. Conocimientos de aplicativos de software libre: Validando el nivel de conocimientos de los encuestados a nivel de conceptos de ingeniería de software y software libre, se encontró que el (52%) de los encuestados está de acuerdo en conocer que es la ingeniería de software y el (53%) afirmó estar muy de acuerdo en que es el software libre, los resultados se pueden validar en la Tabla 10 y la Tabla 11 y visualizar en la Figura 10 y Figura 11.

Tabla 10. Resultados pregunta 1

P1. Entiendo el concepto de ingeniería de software	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	5	6
Ni de acuerdo ni en desacuerdo	14	16
De acuerdo	46	52
Muy de acuerdo	22	25

Fuente. El autor

Figura 10. Porcentajes de resultados de la pregunta 1



Fuente. El autor

Tabla 11. Resultados pregunta 2

P2. Entiendo el concepto de software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	0	0
Ni de acuerdo ni en desacuerdo	13	15
De acuerdo	27	31
Muy de acuerdo	47	53

Fuente. El autor

Figura 11. Porcentajes de resultados de la pregunta 2



Fuente. El autor

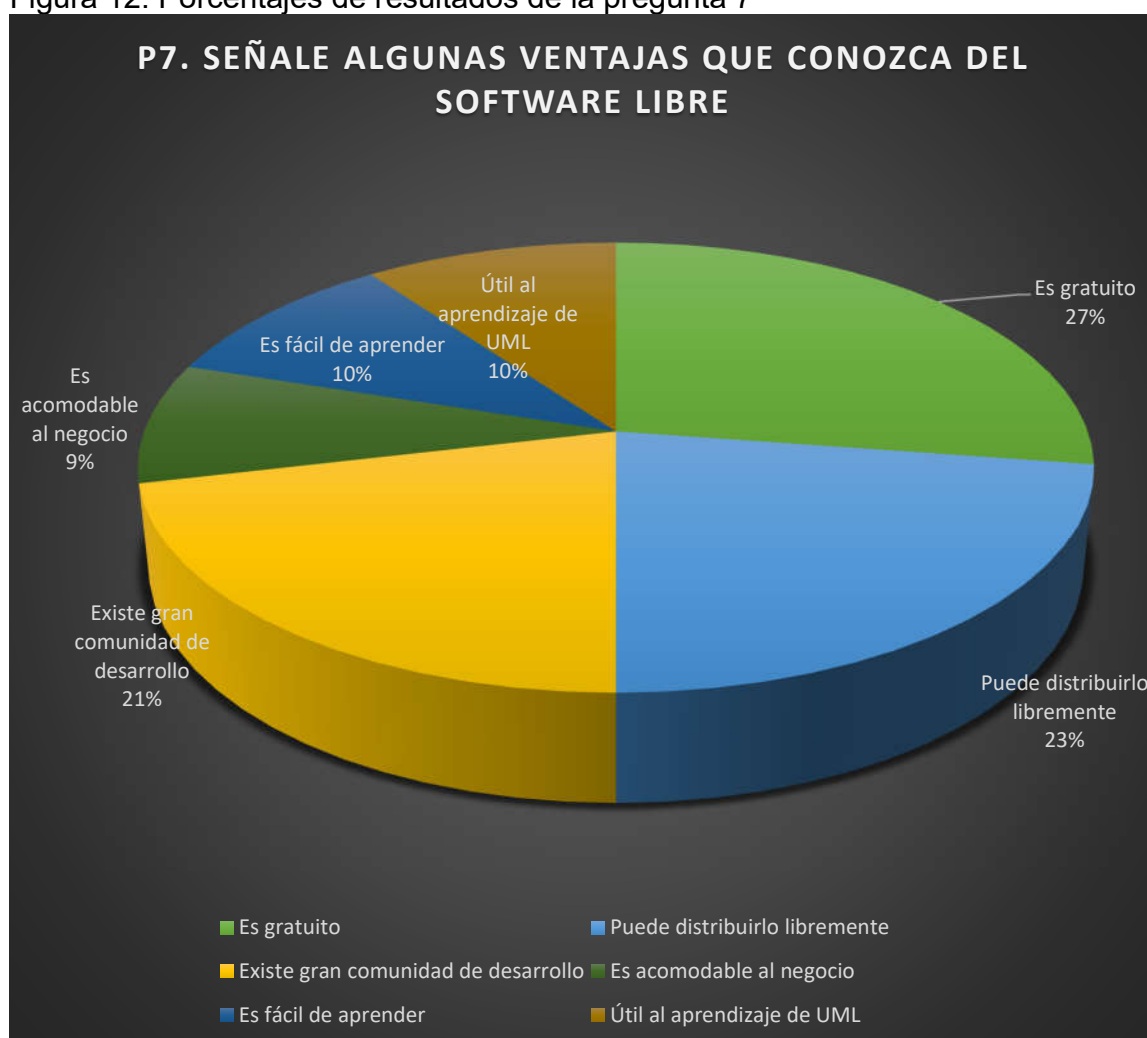
Con respecto al conocimiento de ventajas del software libre se evidencia en que un (27%) de los encuestados considera importante que sea gratuito y un (23%) que se puede distribuir libremente, analizando los resultados de la Tabla 12 se puede notar que los estudiantes encuentran casi por igual las ventajas del software libre identificándolas en su totalidad como se muestra en la Figura 12. Con base a esto se identifica que el (37%) de los estudiantes conoce la herramienta StarUML seguida de un (17%) de ArgoUML, estas dos herramientas son comúnmente utilizadas por su facilidad de uso y aplicación en el área de diseño en ingeniería de software, es importante anotar que un (10%) de los estudiantes no conoce ninguna herramienta de software libre para manejo de UML, los resultados se muestra en la Tabla 13 y la Figura 13.

Tabla 12. Resultados pregunta 7

P7. Señale algunas ventajas que conozca del software libre	CANTIDAD	PORCENTAJE
Es gratuito	82	27
Puede distribuirlo libremente	68	23
Existe gran comunidad de desarrollo	64	21
Es acomodable al negocio	26	9
Es fácil de aprender	30	10
Útil al aprendizaje de UML	30	10

Fuente. El autor

Figura 12. Porcentajes de resultados de la pregunta 7



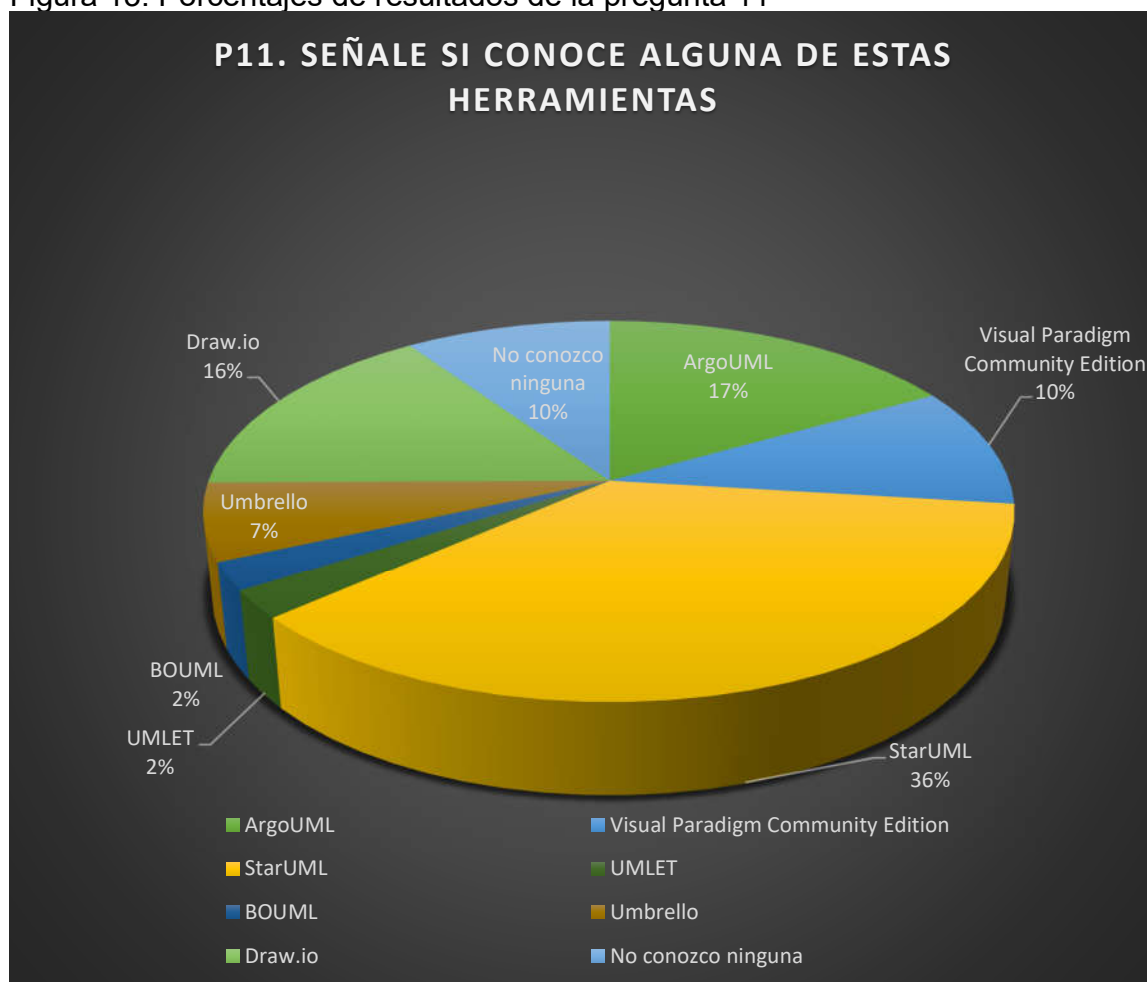
Fuente. El autor

Tabla 13. Resultados pregunta 11

P11. Señale si conoce alguna de estas herramientas	CANTIDAD	PORCENTAJE
ArgoUML	28	17
Visual Paradigm Community Edition	17	10
StarUML	61	37
UMLET	4	2
BOUML	4	2
Umbrello	11	7
Draw.io	26	16
No conozco ninguna	16	10

Fuente. El autor

Figura 13. Porcentajes de resultados de la pregunta 11



Fuente. El autor

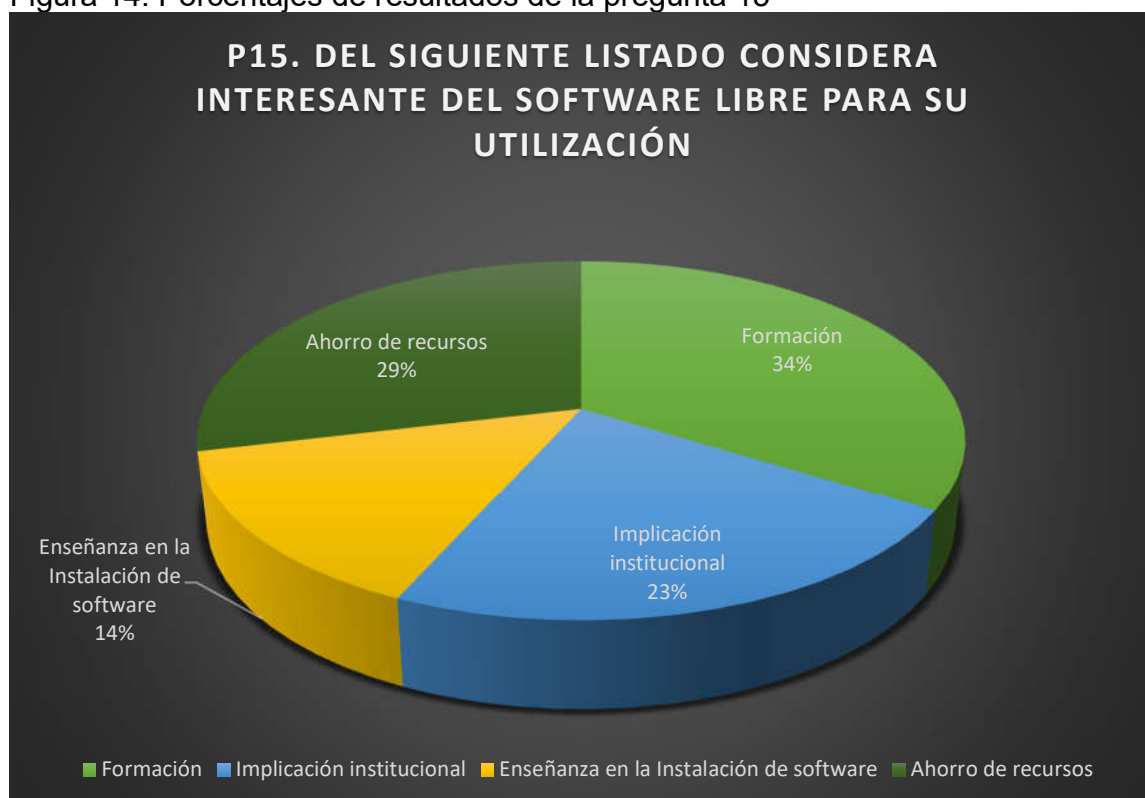
Por último, en conocimiento de los encuestados se identifica que la característica más importante con un (34%) es la formación en el uso de las herramientas de software libre, como se evidencia en la Tabla 14 y Figura 14, asociado a esto se identifica con un (47%) que afirman estar muy de acuerdo con la necesidad de adquirir más conocimientos aplicados a la formación como se muestra en la Tabla 15 y Figura 15.

Tabla 14. Resultados pregunta 15

P15. Del siguiente listado considera interesante del software libre para su utilización	CANTIDAD	PORCENTAJE
Formación	79	34
Implicación institucional	53	23
Enseñanza en la Instalación de software	34	15
Ahorro de recursos	67	29

Fuente. El autor

Figura 14. Porcentajes de resultados de la pregunta 15



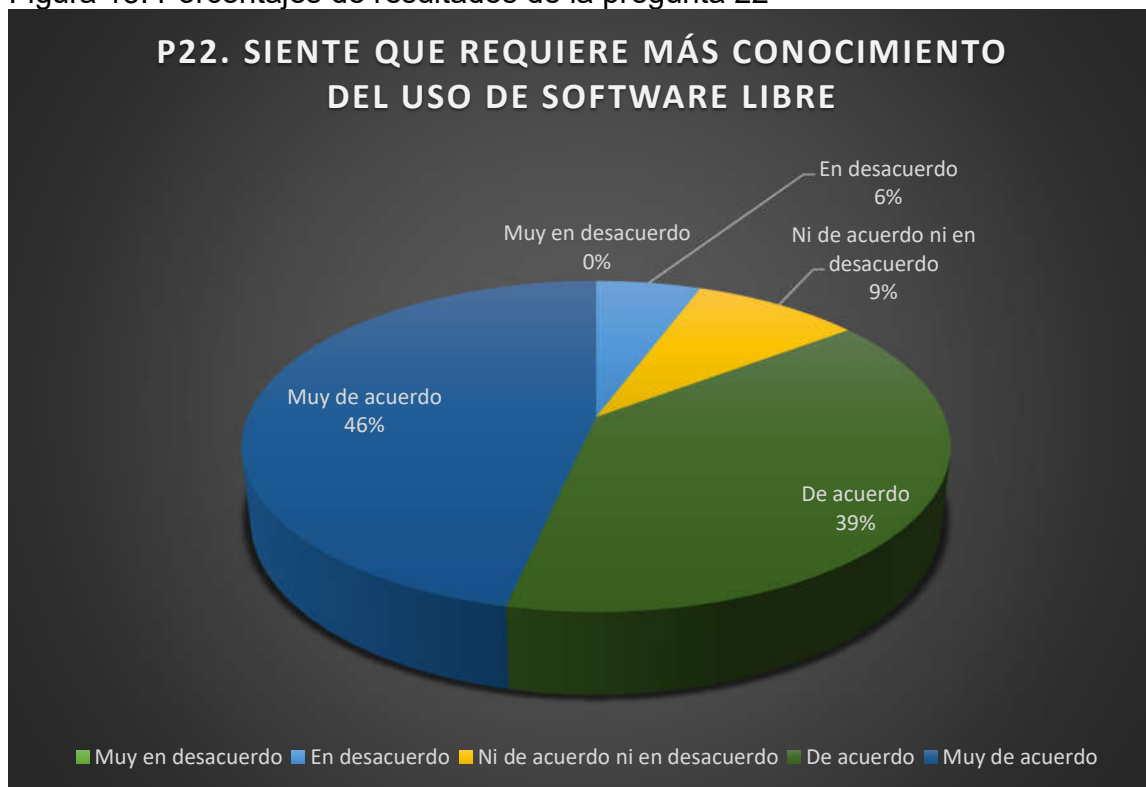
Fuente. El autor

Tabla 15. Resultados pregunta 22

P22. Siente que requiere más conocimiento del uso de software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	0	0
En desacuerdo	5	6
Ni de acuerdo ni en desacuerdo	8	9
De acuerdo	34	39
Muy de acuerdo	41	47

Fuente. El autor

Figura 15. Porcentajes de resultados de la pregunta 22



Fuente. El autor

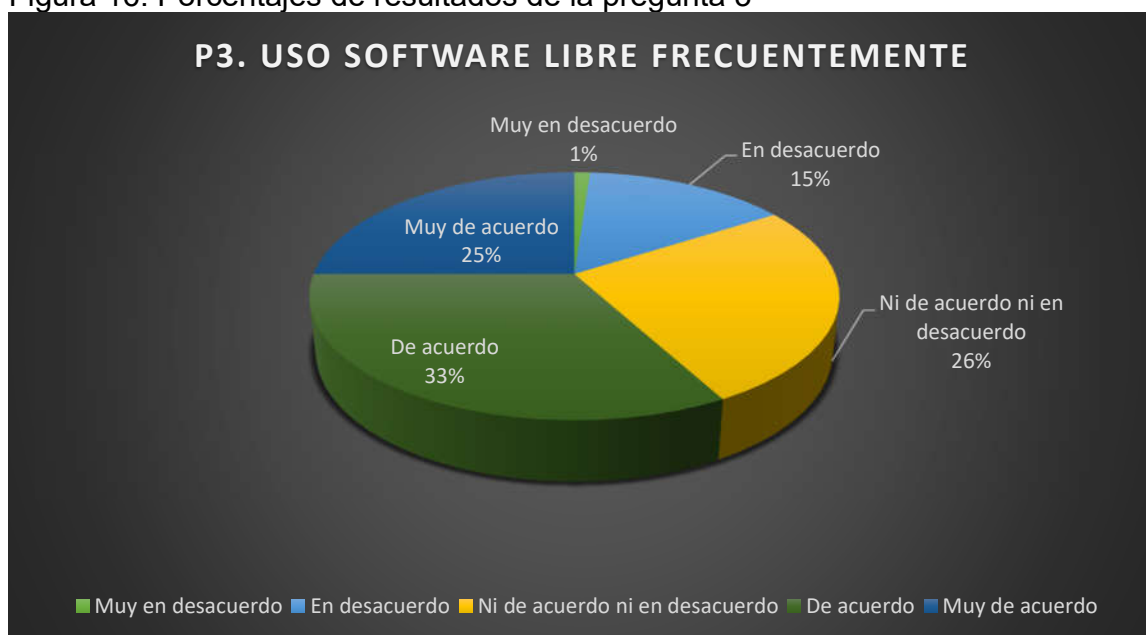
7.2.1.4. Uso de aplicativos de software libre en general: En este apartado se evaluó como los estudiantes perciben el uso de software libre, en el caso de la frecuencia se encontró un (33%) de acuerdo, visto en la Tabla 16 y Figura 16, lo anterior va de la mano con un (33%) en el uso de algún tipo de software libre evidenciando que es poco el uso de aplicativos UML de software libre, como lo muestra la Tabla 17 y Figura 17.

Tabla 16. Resultados pregunta 3

P3. Uso software libre frecuentemente	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	13	15
Ni de acuerdo ni en desacuerdo	23	26
De acuerdo	29	33
Muy de acuerdo	22	25

Fuente. El autor

Figura 16. Porcentajes de resultados de la pregunta 3



Fuente. El autor

Tabla 17. Resultados pregunta 5

P5. Utilicé algún software en el diseño de UML	CANTIDAD	PORCENTAJE
Muy en desacuerdo	11	13
En desacuerdo	9	10
Ni de acuerdo ni en desacuerdo	13	15
De acuerdo	29	33
Muy de acuerdo	26	30

Fuente. El autor

Figura 17. Porcentajes de resultados de la pregunta 5



Fuente. El autor

Desde otro punto de vista se muestra en la Tabla 18 y Figura 18 que un (67%) de los estudiantes si ha utilizado software libre en la fase de diseño UML, pero por otra parte en la Tabla 19 y Figura 19 se evidencia que no hay una claridad en la preferencia de uso de software libre, un (47%) no está ni de acuerdo ni en desacuerdo con la utilización.

Tabla 18. Resultados pregunta 14

P14. En algún momento ha utilizado software libre en el diseño de diagramas UML	CANTIDAD	PORCENTAJE
SI	59	67
NO	29	33

Fuente. El autor

Figura 18. Porcentajes de resultados de la pregunta 14



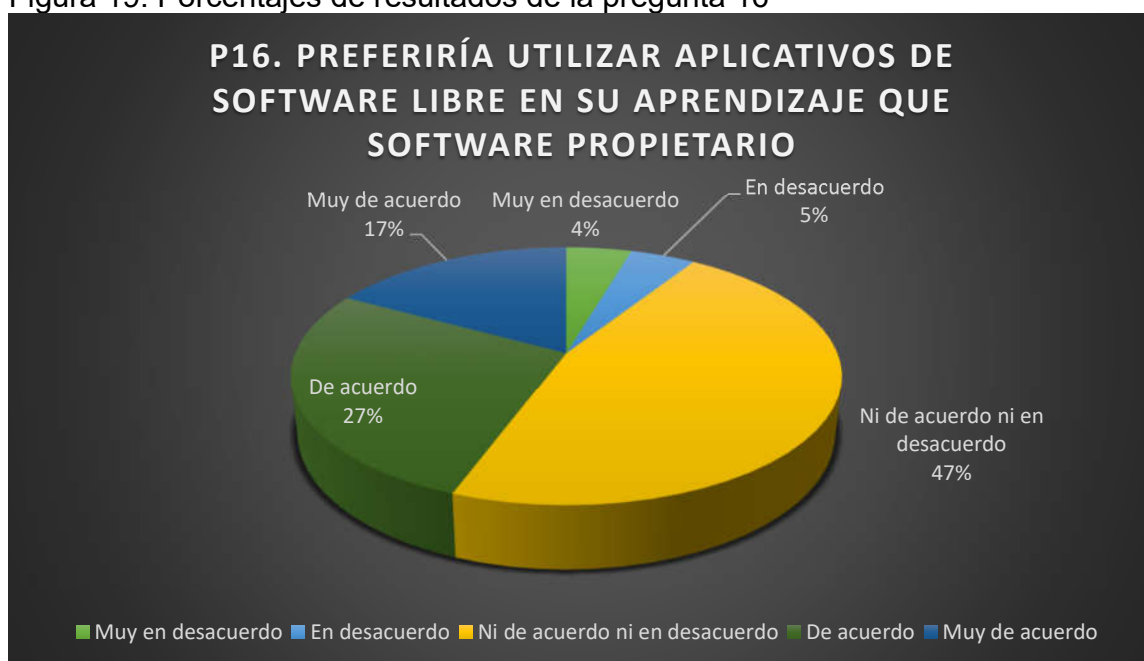
Fuente. El autor

Tabla 19. Resultados pregunta 16

P16. Preferiría utilizar aplicativos de software libre en su aprendizaje que software propietario	CANTIDAD	PORCENTAJE
Muy en desacuerdo	4	5
En desacuerdo	4	5
Ni de acuerdo ni en desacuerdo	41	47
De acuerdo	24	27
Muy de acuerdo	15	17

Fuente. El autor

Figura 19. Porcentajes de resultados de la pregunta 16



Fuente. El autor

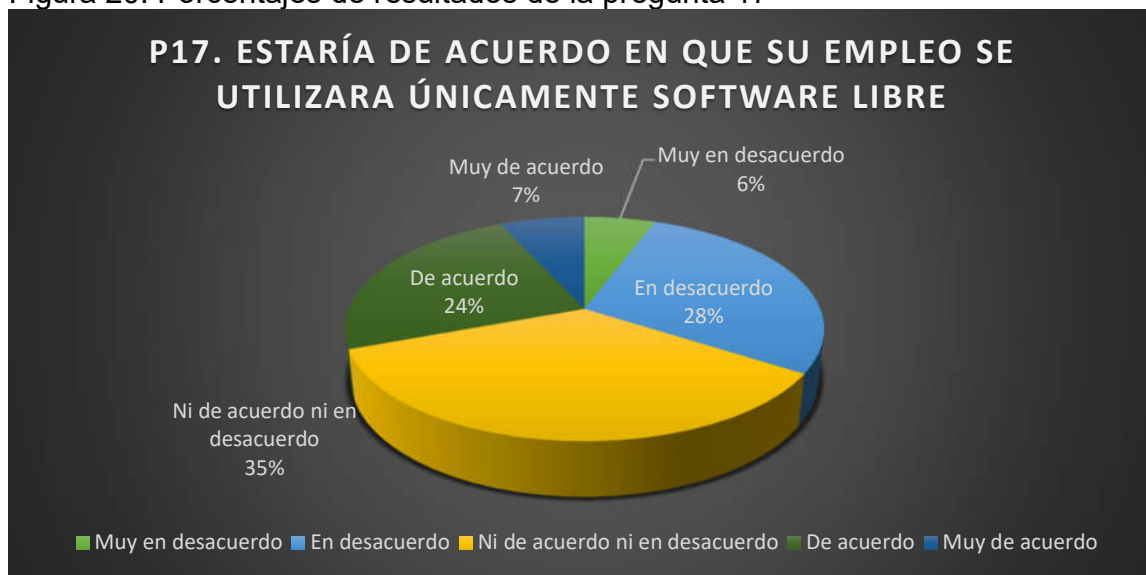
Desde un punto de vista laboral actual y a futuro se muestra que los encuestados pueden llegar o no a utilizar herramientas de software libre, un (35%) muestra su posición con respecto a esto donde se ve en la Tabla 20 y Figura 20. Con relación a lo anterior, se encuentra un (51%) de acuerdo sobre la utilidad en el uso de software libre para la enseñanza y el (35%) muy de acuerdo como lo muestra la Tabla 21 y Figura 21.

Tabla 20. Resultados pregunta 17

P17. Estaría de acuerdo en que su empleo se utilizara únicamente software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	5	6
En desacuerdo	25	28
Ni de acuerdo ni en desacuerdo	31	35
De acuerdo	21	24
Muy de acuerdo	6	7

Fuente. El autor

Figura 20. Porcentajes de resultados de la pregunta 17



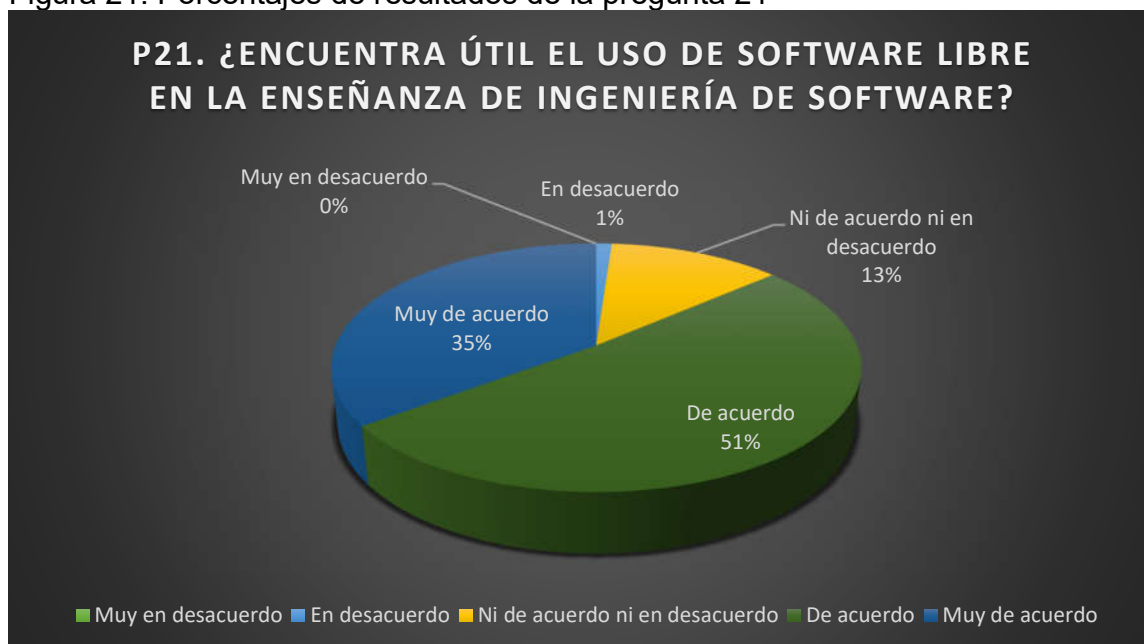
Fuente. El autor

Tabla 21. Resultados pregunta 21

P21. ¿Encuentra útil el uso de software libre en la enseñanza de ingeniería de software?	CANTIDAD	PORCENTAJE
Muy en desacuerdo	0	0
En desacuerdo	1	1
Ni de acuerdo ni en desacuerdo	11	13
De acuerdo	45	51
Muy de acuerdo	31	35

Fuente. El autor

Figura 21. Porcentajes de resultados de la pregunta 21



Fuente. El autor

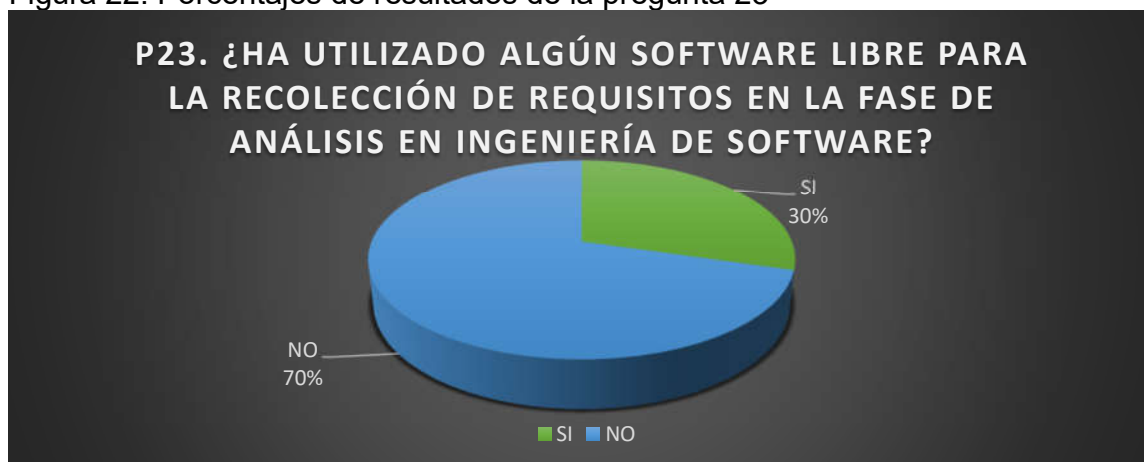
Con base al uso enfocado específicamente en ingeniería de software se evidencia que un (30%) a utilizado algún tipo de herramienta de software libre en la fase de análisis y un (67%) en la fase de diseño, lo anterior hace referencia al mayor desarrollo de diagramas de la fase de diseño contra la de análisis, estos datos son mostrados en la Tabla 23 y Tabla 24 y representados en la Figura 22 y Figura 23. Este análisis es importante desde el punto de vista de preferencia hacia el software libre.

Tabla 22. Resultados pregunta 23

P23. ¿Ha utilizado algún software libre para la recolección de requisitos en la fase de análisis en ingeniería de software?	CANTIDAD	PORCENTAJE
SI	26	30
NO	62	70

Fuente. El autor

Figura 22. Porcentajes de resultados de la pregunta 23



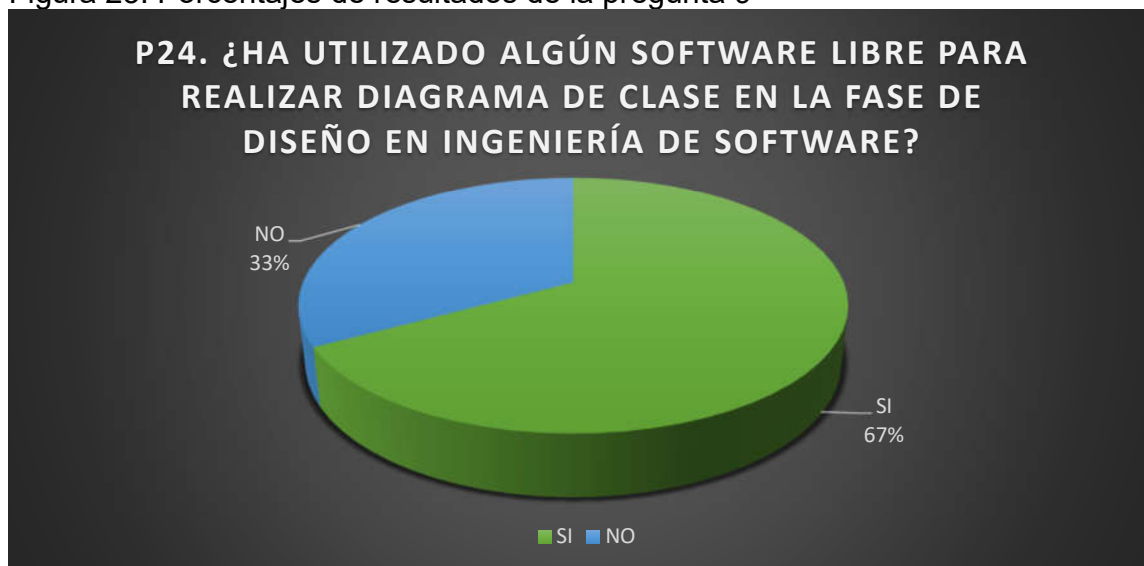
Fuente. El autor

Tabla 23. Resultados pregunta 24

P24. ¿Ha utilizado algún software libre para realizar diagrama de clase en la fase de diseño en ingeniería de software?	CANTIDAD	PORCENTAJE
SI	59	67
NO	29	33

Fuente. El autor

Figura 23. Porcentajes de resultados de la pregunta 9



Fuente. El autor

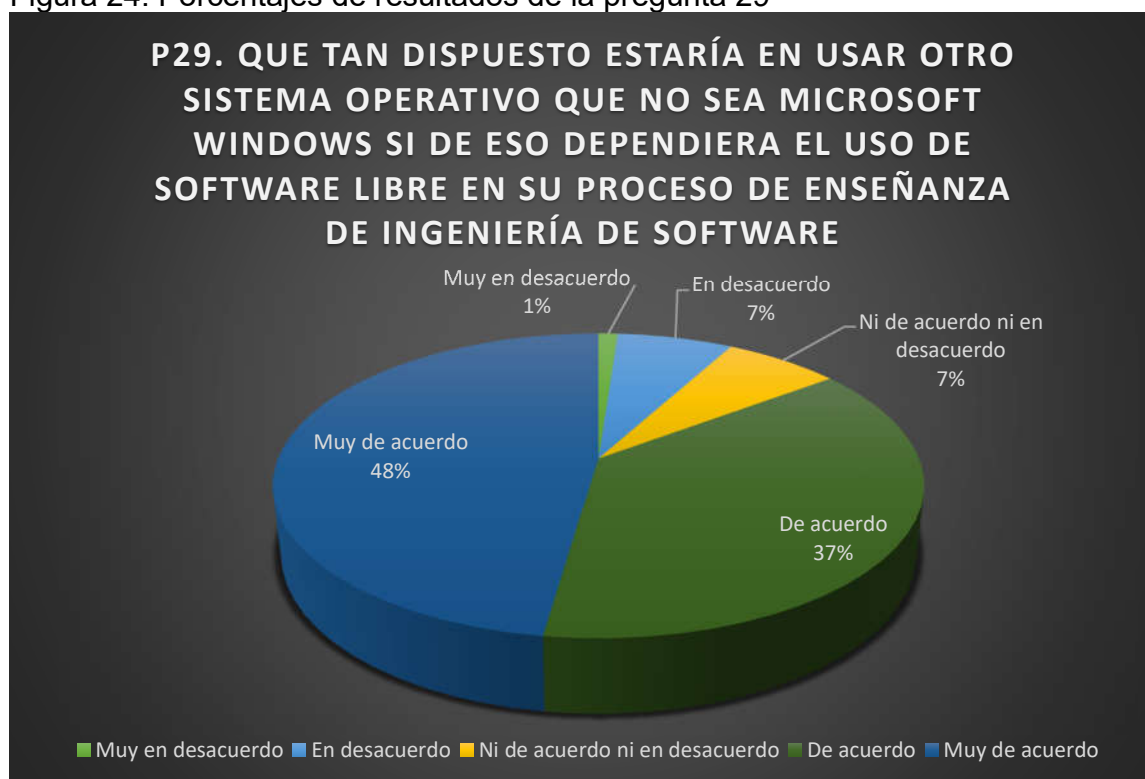
En cuanto a disposición en el uso de software libre se evidencia que un (48%) de los encuestados están muy de acuerdo y un (38%) de acuerdo, lo que suma un (86%) de aceptación en la enseñanza de ingeniería de software utilizando software libre. Los resultados se muestran en la Tabla 24 y Figura 24.

Tabla 24. Resultados pregunta 29

P29. Que tan dispuesto estaría en usar otro sistema operativo que no sea Microsoft Windows si de eso dependiera el uso de software libre en su proceso de enseñanza de ingeniería de software	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	6	7
Ni de acuerdo ni en desacuerdo	6	7
De acuerdo	33	38
Muy de acuerdo	42	48

Fuente. El autor

Figura 24. Porcentajes de resultados de la pregunta 29



Fuente. El autor

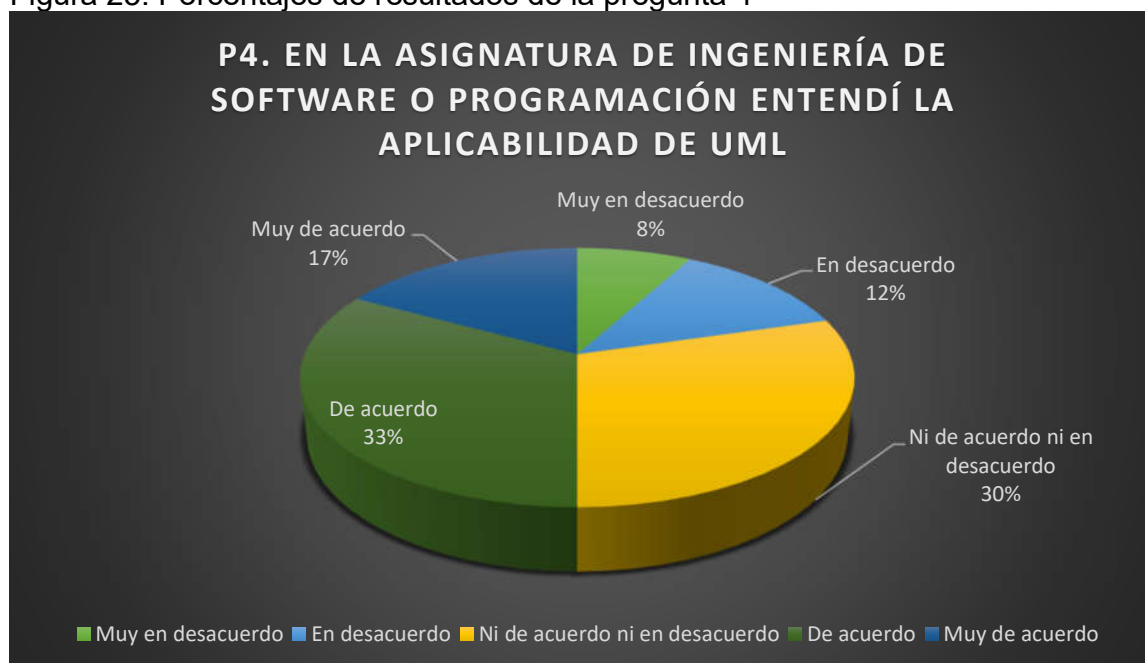
7.2.1.5. Nivel de satisfacción del uso de software libre para aprender UML: En este apartado se evalúa el nivel de inclinación o no específico en herramientas de software libre para UML, en el primer caso visto en la Tabla 25 y Figura 25 se muestra un (33%) de acuerdo en que se entendió la aplicabilidad de UML en ingeniería de software, es importante tener en cuenta este elemento ya que es base para verificar el uso o no de software libre, además se identifica un (69%) en que se está muy de acuerdo en el uso de software libre para la enseñanza evidenciado en la Tabla 26 y representado en la Figura 26.

Tabla 25. Resultados pregunta 4

P4. En la asignatura de ingeniería de software o programación entendí la aplicabilidad de UML	CANTIDAD	PORCENTAJE
Muy en desacuerdo	7	8
En desacuerdo	11	13
Ni de acuerdo ni en desacuerdo	26	30
De acuerdo	29	33
Muy de acuerdo	15	17

Fuente. El autor

Figura 25. Porcentajes de resultados de la pregunta 4



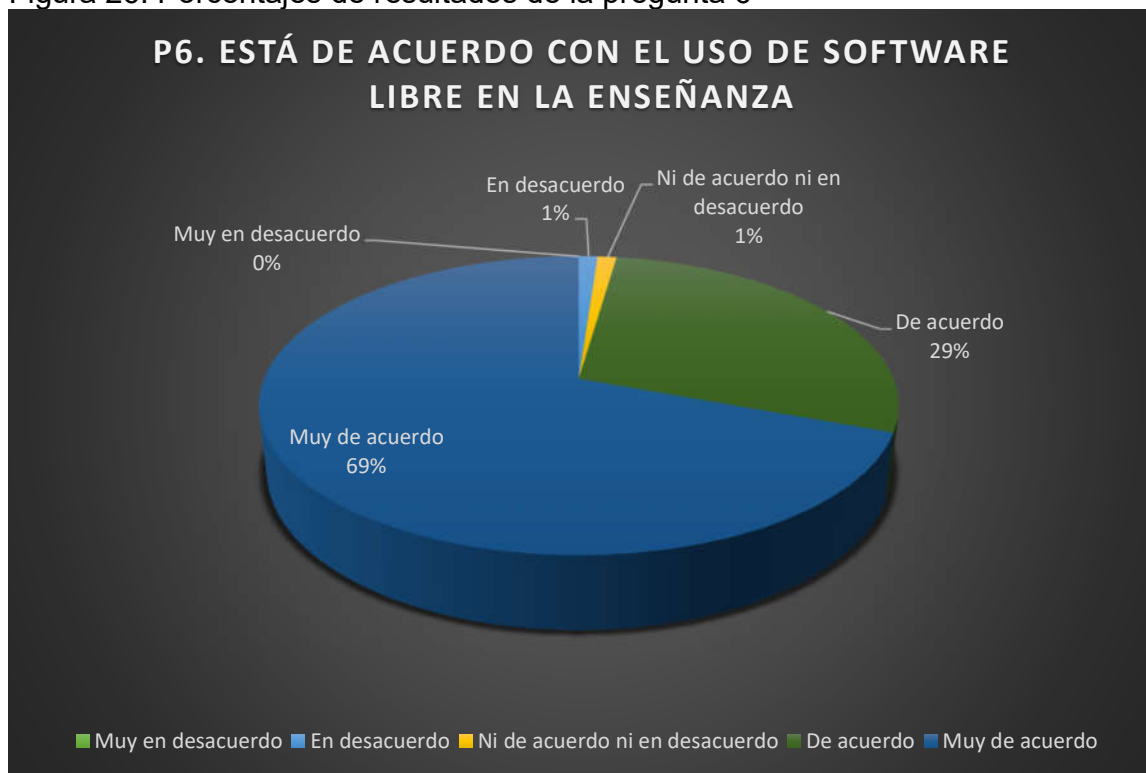
Fuente. El autor

Tabla 26. Resultados pregunta 6

P6. Está de acuerdo con el uso de software libre en la enseñanza	CANTIDAD	PORCENTAJE
Muy en desacuerdo	0	0
En desacuerdo	1	1
Ni de acuerdo ni en desacuerdo	1	1
De acuerdo	25	28
Muy de acuerdo	61	69

Fuente. El autor

Figura 26. Porcentajes de resultados de la pregunta 6



Fuente. El autor

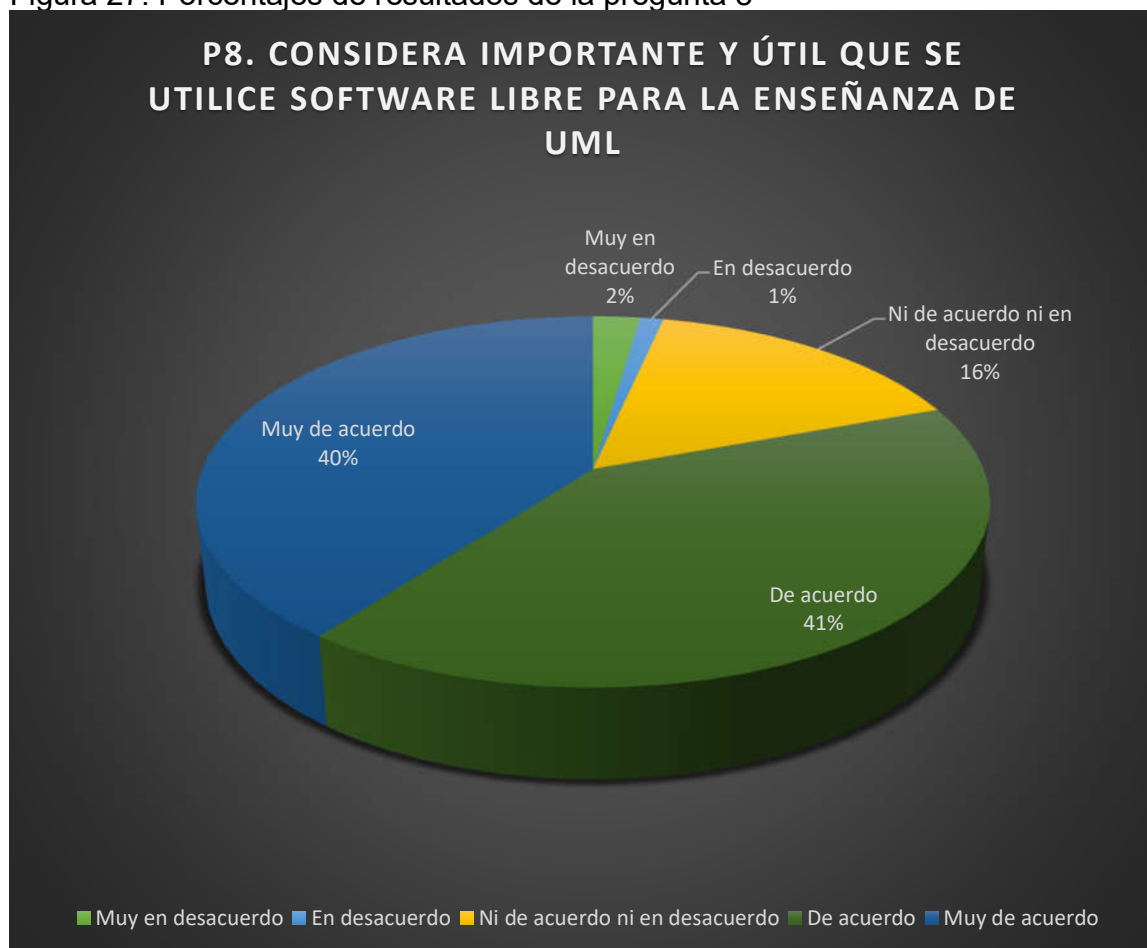
Otro punto a tener en cuenta es la importancia en la utilización de software libre en la enseñanza específica de UML y verificar si se está o no de acuerdo en que la institución de educación superior implemente este tipo de soluciones, el (41%) de los encuestados manifiesta estar de acuerdo con que se utilice software libre y el (45%) muy de acuerdo más un (42%) de acuerdo lo que suma un (87%) estar a favor en el uso de software libre en la institución. Las Tabla 27 y Tabla 28 muestran los porcentajes y la Figura 27 y Figura 28 su representación.

Tabla 27. Resultados pregunta 8

P8. Considera importante y útil que se utilice software libre para la enseñanza de UML	CANTIDAD	PORCENTAJE
Muy en desacuerdo	2	2
En desacuerdo	1	1
Ni de acuerdo ni en desacuerdo	14	16
De acuerdo	36	41
Muy de acuerdo	35	40

Fuente. El autor

Figura 27. Porcentajes de resultados de la pregunta 8



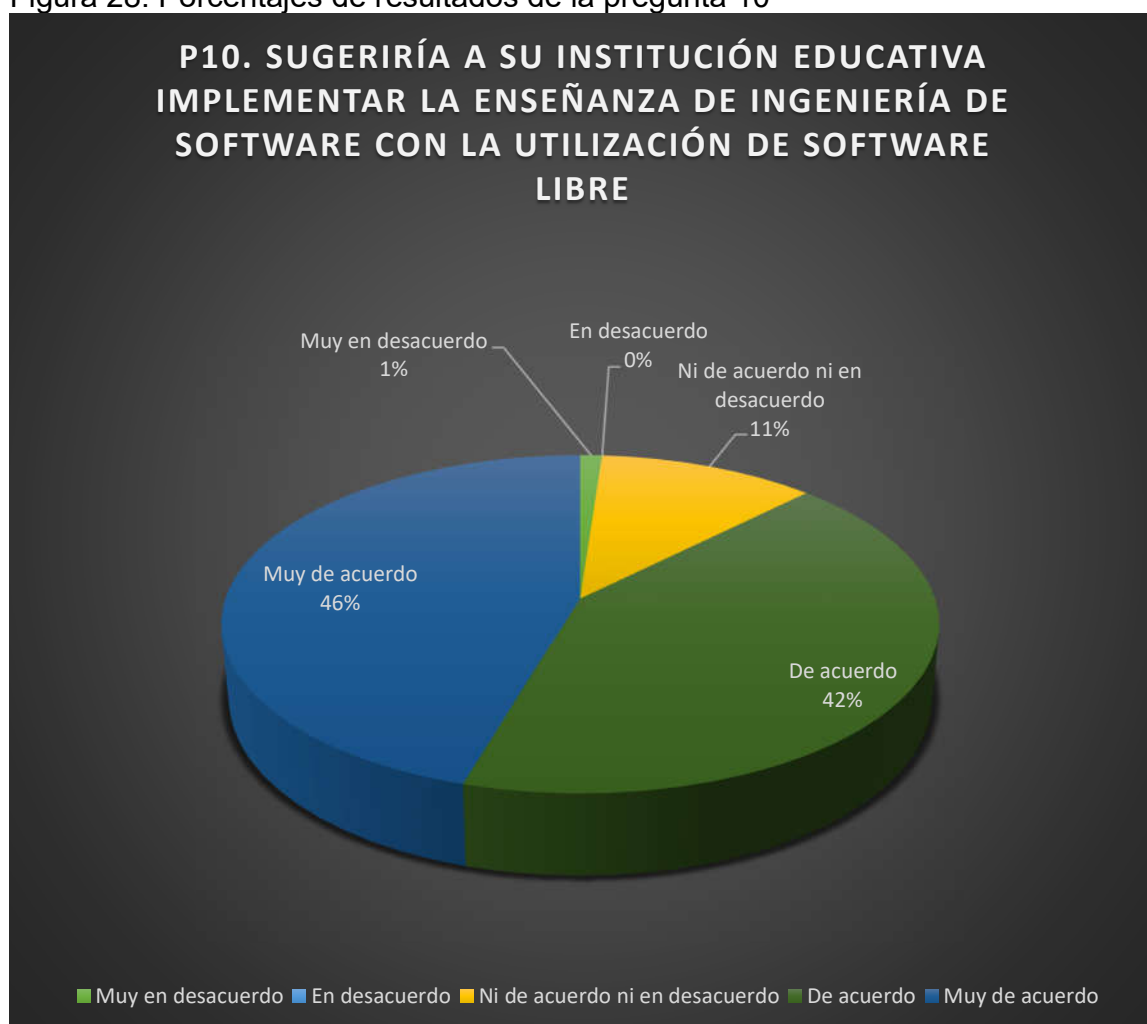
Fuente. El autor

Tabla 28. Resultados pregunta 10

P10. Sugeriría a su institución educativa implementar la enseñanza de ingeniería de software con la utilización de software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	1	1
En desacuerdo	0	0
Ni de acuerdo ni en desacuerdo	10	11
De acuerdo	37	42
Muy de acuerdo	40	45

Fuente. El autor

Figura 28. Porcentajes de resultados de la pregunta 10



Fuente. El autor

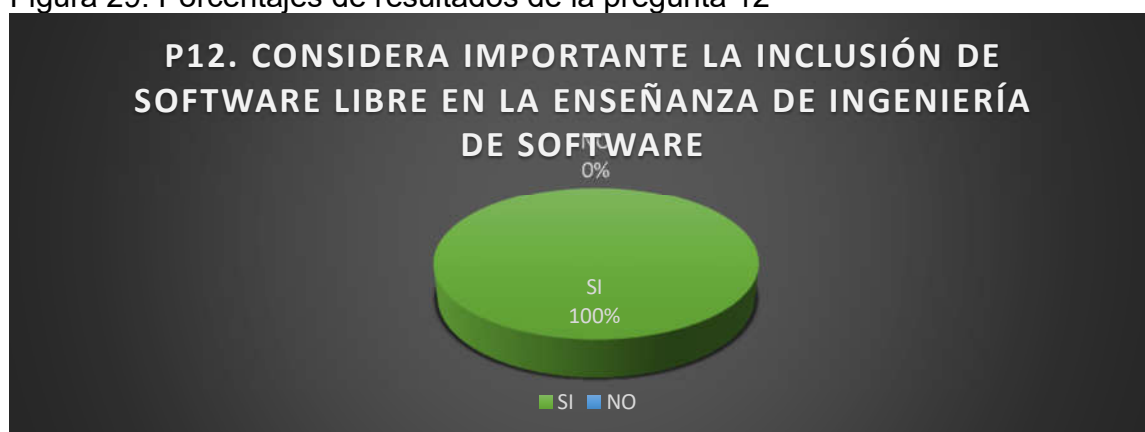
Luego de validar los niveles de satisfacción a nivel de institución educativa se hace necesario validar si la inclusión de software libre es aceptada o no y si la calidad en el proceso de formación educativa se vería afectado, un (100%) de la población encuestado ve favorable la inclusión de software libre en la enseñanza de ingeniería de software y un (34%) está muy en desacuerdo en que su proceso de educación se vería afectado si se incorpora software libre en su enseñanza. Los resultados se muestran en la Tabla 29 y Tabla 30 y en la Figura 29 y Figura 30.

Tabla 29. Resultados pregunta 12

P12. Considera importante la inclusión de software libre en la enseñanza de ingeniería de software	CANTIDAD	PORCENTAJE
SI	88	100
NO	0	0

Fuente. El autor

Figura 29. Porcentajes de resultados de la pregunta 12



Fuente. El autor

Tabla 30. Resultados pregunta 13

P13. Su nivel de formación se vería afectado si se enseñara con software libre	CANTIDAD	PORCENTAJE
Muy en desacuerdo	30	34
En desacuerdo	14	16
Ni de acuerdo ni en desacuerdo	24	27
De acuerdo	16	18
Muy de acuerdo	4	5

Fuente. El autor

Figura 30. Porcentajes de resultados de la pregunta 13



Fuente. El autor

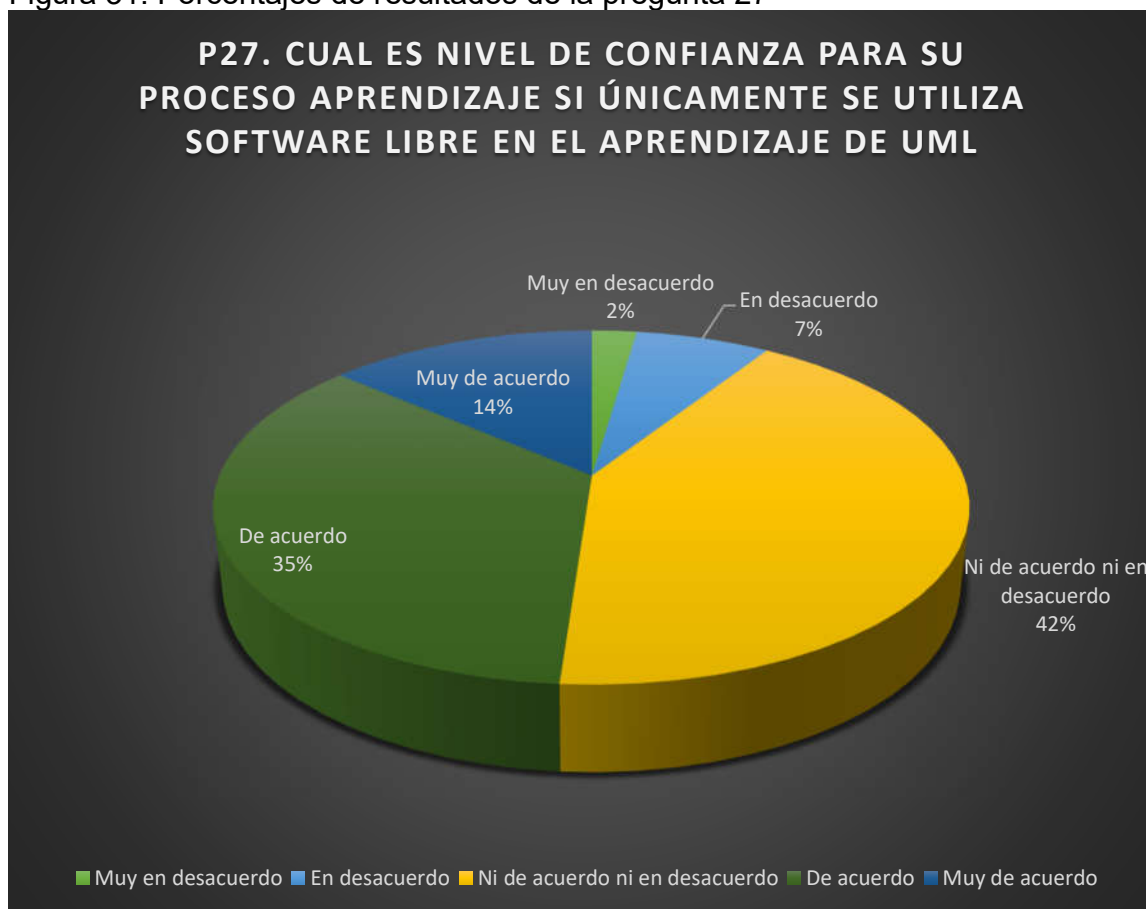
Por último, se valida el nivel de confianza, opinión en la mezcla de software libre y propietario y los niveles de satisfacción en cuanto a usar herramientas de software libre que tienen menos herramientas que las privativas. En el análisis se identificó que un (42%) no está de acuerdo ni en desacuerdo en que únicamente se use software libre, pero un (35%) estaría de acuerdo lo que hace que exista ese nivel de confianza deseado, por otra parte, el (40%) estaría de acuerdo en la mezcla de aplicativos de enseñanza y un (42%) en el uso de herramientas de software libre con las limitantes que algunas tienen. Los resultados se muestran en la Tabla 31, Tabla 32 y Tabla 33 y se representan en la Figura 31, Figura 32 y Figura 33

Tabla 31. Resultados pregunta 27

P27. Cual es nivel de confianza para su proceso aprendizaje si únicamente se utiliza software libre en el aprendizaje de UML	CANTIDAD	PORCENTAJE
Muy en desacuerdo	2	2
En desacuerdo	6	7
Ni de acuerdo ni en desacuerdo	37	42
De acuerdo	31	35
Muy de acuerdo	12	14

Fuente. El autor

Figura 31. Porcentajes de resultados de la pregunta 27



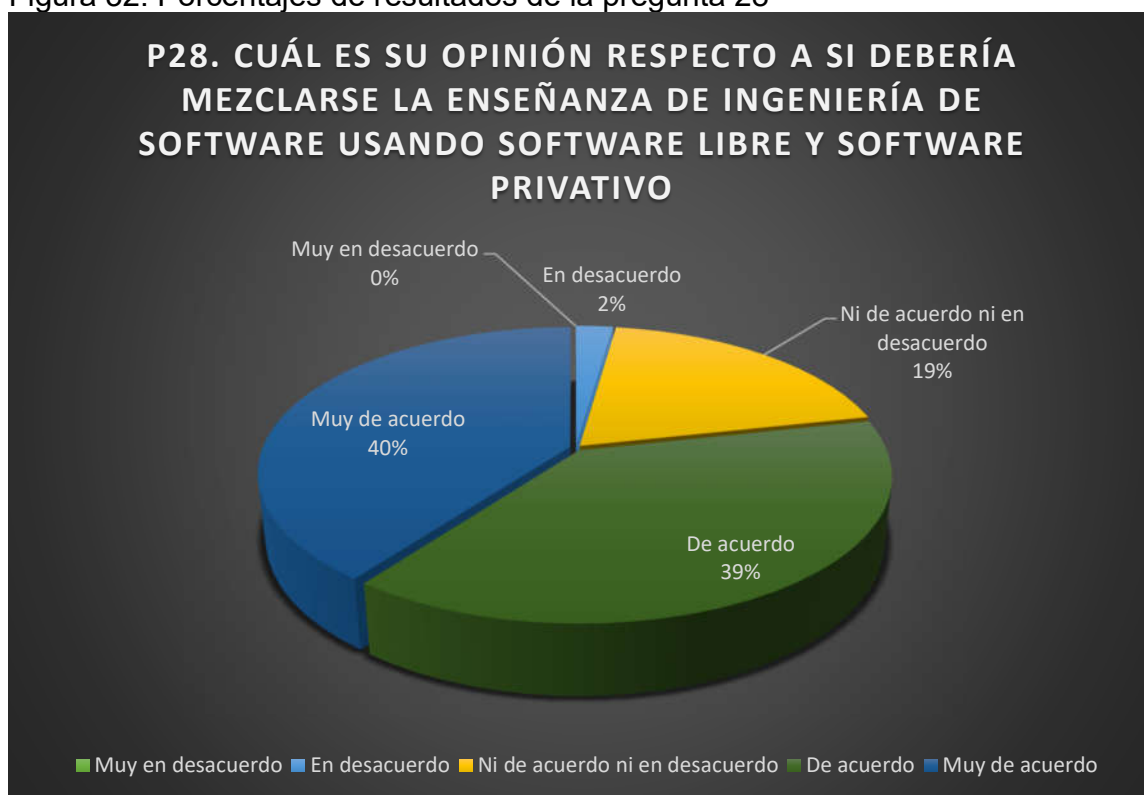
Fuente. El autor

Tabla 32. Resultados pregunta 28

P28.Cuál es su opinión respecto a si debería mezclarse la enseñanza de ingeniería de software usando software libre y software privativo	CANTIDAD	PORCENTAJE
Muy en desacuerdo	0	0
En desacuerdo	2	2
Ni de acuerdo ni en desacuerdo	17	19
De acuerdo	34	39
Muy de acuerdo	35	40

Fuente. El autor

Figura 32. Porcentajes de resultados de la pregunta 28



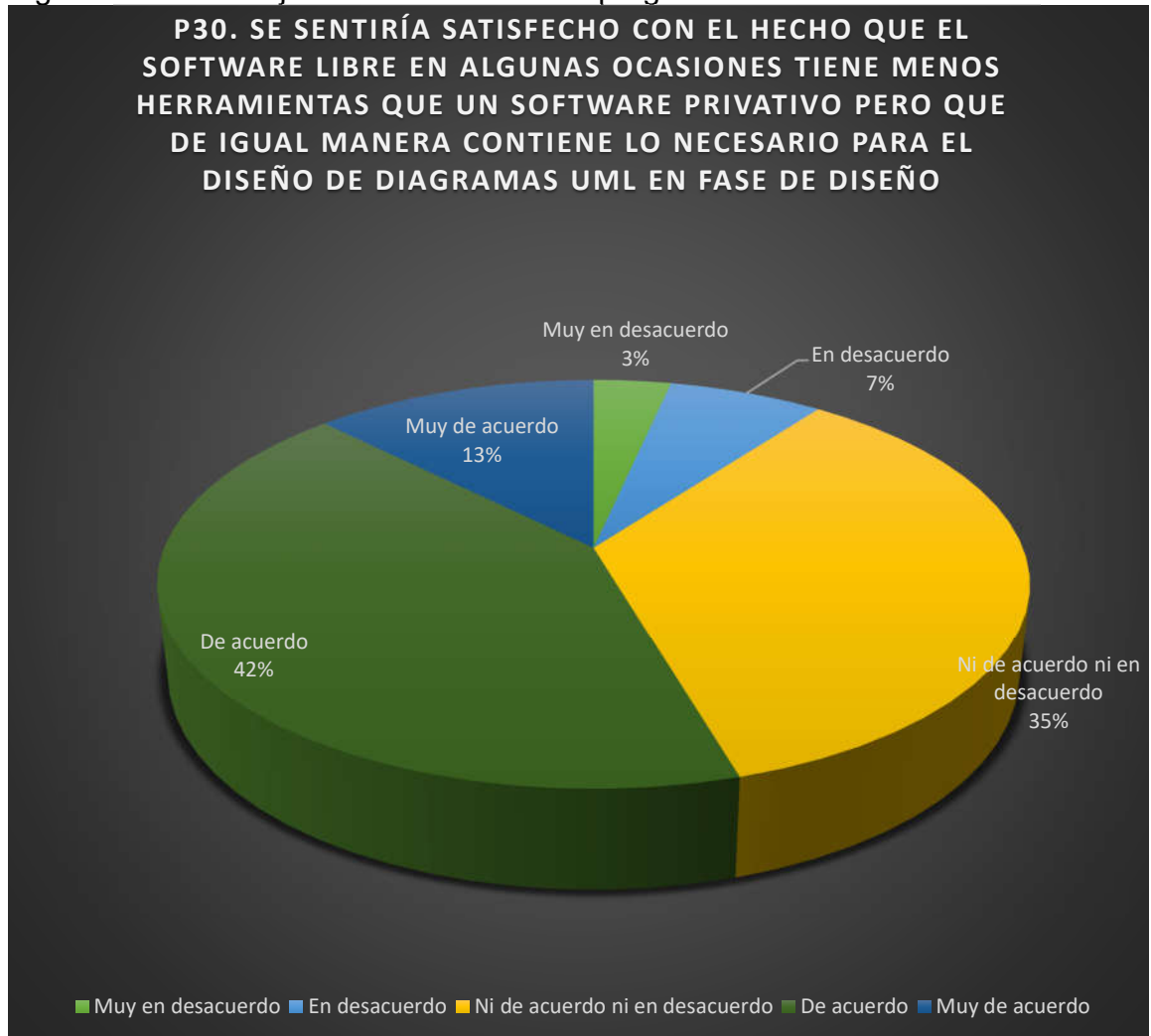
Fuente. El autor

Tabla 33. Resultados pregunta 30

P30. Se sentiría satisfecho con el hecho que el software libre en algunas ocasiones tiene menos herramientas que un software privativo pero que de igual manera contiene lo necesario para el diseño de diagramas UML en fase de diseño	CANTIDAD	PORCENTAJE
Muy en desacuerdo	3	3
En desacuerdo	6	7
Ni de acuerdo ni en desacuerdo	31	35
De acuerdo	37	42
Muy de acuerdo	11	13

Fuente. El autor

Figura 33. Porcentajes de resultados de la pregunta 30



Fuente. El autor

7.2.2. Análisis de resultados: Con base a los resultados obtenidos se evaluaron las 4 variables principales del estudio según los porcentajes identificados, las 4 variables son: Nivel de satisfacción del estudiante, conocimientos de aplicativos de software libre, uso de aplicativos de software libre en general, nivel de satisfacción del uso de software libre para aprender UML.

Con base en esto se presentan los resultados del análisis en la Tabla 34. *Resultados globales.*

Tabla 34. Resultados globales

VARIABLE	PREGUNTA	PORCENTAJE	RESULTADO
Nivel de satisfacción del estudiante	P9 - ¿Considera su formación adecuada con el uso de software libre?	45	De acuerdo
	P18 - ¿Está de acuerdo con las políticas de su institución educativa en cuanto al uso de software libre?	52	Ni de acuerdo, ni en desacuerdo
	P19 - Desde su punto de vista, ¿El nivel de equipos de cómputo de la institución permitiría la inclusión de software libre?	35	De acuerdo
	P20 - ¿El nivel de apropiación de sus docentes en la enseñanza de software libre es el adecuado?	39	De acuerdo
	P25 - ¿Se ha sentido a gusto usando software libre?	43	De acuerdo
	P26 - ¿Que tanto recomendaría el uso de software libre en su aprendizaje?	57	De acuerdo
Conocimientos de aplicativos de software libre	P1 - ¿Entiende el concepto de ingeniería de software?	52	De acuerdo
	P2 - ¿Entiendo el concepto de software libre?	53	Muy de acuerdo
	P7 - Señale algunas ventajas que conozca del software libre UML	27	Gratuita
	P11 - Señale si conoce alguna de estas herramientas	37	StarUML
	P15 - Del siguiente listado, ¿Considera interesante del software libre para su utilización?	34	Formación
	P22 - ¿Siente que requiere más conocimiento del uso de software libre?	47	Muy de acuerdo
Uso de aplicativos de software libre en general	P3 - ¿Usa software libre frecuentemente?	33	De acuerdo
	P5 - ¿Utiliza algún software en el diseño de UML?	33	De acuerdo
	P14 - ¿En algún momento ha utilizado software libre en el diseño de diagramas UML?	67	SI

VARIABLE	PREGUNTA	PORCENTAJE	RESULTADO
	P16 - ¿Preferiría utilizar aplicativos de software libre en su aprendizaje que software propietario?	47	Ni de acuerdo, ni en desacuerdo
	P17 - ¿Estaría de acuerdo en que su empleo se utilizara únicamente software libre?	35	Ni de acuerdo, ni en desacuerdo
	P21 - ¿Encuentra útil el uso de software libre en la enseñanza de ingeniería de software?	51	De acuerdo
	P23 - ¿Ha utilizado algún software libre para la recolección de requisitos en la fase de análisis en ingeniería de software?	70	NO
	P24 - ¿Ha utilizado algún software libre para realizar diagrama de clase en la fase de diseño en ingeniería de software?	67	SI
	P29 - ¿Que tan dispuesto estaría en usar otro sistema operativo que no sea Microsoft Windows si de eso dependiera el uso de software libre en su proceso de enseñanza de ingeniería de software?	48	Muy de acuerdo
Nivel de satisfacción del uso de software libre para aprender UML	P4 - ¿En la asignatura de ingeniería de software o programación entendí la aplicabilidad de UML?	33	De acuerdo
	P6 - ¿Está de acuerdo con el uso de software libre en la enseñanza?	69	Muy de acuerdo
	P8 - ¿Considera importante y útil se utilice software libre en enseñanza de UML?	41	De acuerdo
	P10 - ¿Sugeriría a su institución educativa implementar la enseñanza de ingeniería de software con la utilización de software libre?	45	Muy de acuerdo
	P12 - ¿Considera importante la inclusión de software libre en la enseñanza de ingeniería de software?	100	SI

VARIABLE	PREGUNTA	PORCENTAJE	RESULTADO
	P13 - ¿Su nivel de formación se vería afectado si se enseña con software libre?	34	Muy en desacuerdo
	P27 - ¿Cual es nivel de confianza para su proceso aprendizaje si únicamente se utiliza software libre en el aprendizaje de UML?	42	Ni de acuerdo, ni en desacuerdo
	P28 - ¿Cuál es su opinión respecto a si debería mezclarse la enseñanza de ingeniería de software usando software libre y software privativo?	40	Muy de acuerdo
	P30 - ¿Se sentiría satisfecho con el hecho que el software libre en algunas ocasiones tiene menos herramientas que un software privativo pero que de igual manera contiene lo necesario para el diseño de diagramas UML en fase de diseño?	42	De acuerdo

Fuente. El autor

7.2.2.1. Nivel de satisfacción del estudiante: Se identifica que los encuestados consideran adecuado el uso de software libre, aunque existe un desconocimiento de las políticas que tiene la institución con respecto a su implementación, además es poca la recepción de instalación de este software en la medida que los equipos de cómputo no se adaptan del todo para su uso, a pesar de esto, la satisfacción de uso es alta y existe una gran acogida en enseñar y aprender utilizando software libre

7.2.2.2. Conocimientos de aplicativos de software libre: En base a conocimientos de ingeniería de software y UML, la mitad de la población conoce del tema, se inclinan en que la mayor ventaja del software libre es que sea gratis y la mayor aplicación utilizada es StarUML, es de resaltar que la mayor importancia para el uso del software libre es la formación, no obstante, es requerido un mayor conocimiento en el uso de este tipo de aplicaciones.

7.2.2.3. Uso de aplicativos de software libre en general: En cuanto al uso de aplicativos en general, hay bajo uso de herramientas de software libre en la comunidad educativa, aunque en promedio su uso es alto en la enseñanza de

ingeniería de software, a pesar de esto, existe indiferencia en el uso y empleo de software libre. Por otra parte, se evidencia que no hay uso de software libre en la fase de análisis de ingeniería de software, pero si en el diseño, aunque existe la disposición a en su utilización en la totalidad de los cursos de ingeniería de software.

7.2.2.4. Nivel de satisfacción del uso de software libre para aprender UML: La satisfacción del estudiante viene determinada en la comprensión, uso, apropiación y aceptación a las herramientas de software libre, en el análisis realizado se puede identificar que hay un 100% de aceptación en la importancia de inclusión de software libre para la enseñanza de UML, además, el nivel de formación no se vería afectado, la confianza aumentaría y no existirían inconvenientes en usar de forma combinada con soluciones propietarias.

7.2.3. Discusión y resultados: a partir del análisis de los resultados obtenidos se validan los objetivos del proyecto de la siguiente manera:

En cuanto al diseño y aplicación de métodos de recolección de información se evidenció que la aplicación del cuestionario a los 88 estudiantes reflejo la receptividad e importancia del ejercicio, es de anotar que la gran mayoría de estudiantes vio con mucho entusiasmo la idea principal del proyecto para la inclusión de software libre en sus procesos de enseñanza, además vieron de forma muy asertiva las preguntas y su clasificación, del total de la muestra se obtuvieron el 100% de los resultados. Cabe anotar que el ejercicio hubiera arrojado resultados más interesantes de haber tomado en cuenta varias instituciones educativas a nivel ciudad y país.

En la evaluación de los resultados de los 4 aspectos determinantes, se evidenció un nivel de apropiación y conocimiento medio en el tema, lo anterior soportado en porcentajes entre el 50% en varias preguntas, la población estudiantil acepta y tiene gran acogida por este tipo de software pero el nivel de conocimiento o información sobre el tema aún es poco, esto genera que aunque se puedan incorporar este tipo de medios se debe concientizar sobre su importancia y sobre todo validez en el proceso de aprendizaje, el objetivo se cumplió a satisfacción ya que revelo resultado muy importantes en la aplicación de software libre en la enseñanza de UML.

El análisis de los resultados permitió reflejar tendencias en el uso de software libre específico como StarUML que es el más conocido y usado por los estudiantes, además de identificar el software libre como gratuito, que, aunque tiene esa ventaja no todo el software de este tipo lo es, se debe informar sobre el uso y deberes que la licencia de este tipo de producto permite. Además, a partir del análisis se pudo llegar a definir elementos como uso, satisfacción e importancia que de alguna manera es la base fundamental para la inclusión de software libre, teniendo en cuenta el tema de ingeniería de software donde se aplique.

8. CONCLUSIONES

El trabajo en general respondió a las expectativas planteadas y generó resultados completos en cuanto al objeto de estudio, la aplicación del cuestionario permitió obtener los elementos de análisis requeridos para entender y evaluar las tendencias hacia el uso de software libre en la enseñanza, de esta forma poder responder los interrogantes planteados en este proyecto.

Con base a los resultados obtenidos se puede concluir lo siguiente

- Desde el punto de vista del nivel de satisfacción de los estudiantes se encuentra que un 50% de los encuestados se encuentra a gusto con su formación, políticas de la institución educativa, equipos tecnológicos, apropiación de conocimiento de los docentes y gusto por su uso, lo anterior da a entender que la adopción de este tipo de software está en crecimiento y puede competir con la inclusión de software propietario en las instituciones educativas.
- Con base a los conocimientos, la tendencia es un poco variable en cuanto entendimiento del concepto de ingeniería de software contra software libre ya que se desconocen las ventajas de aplicar metodologías UML por medio de herramientas libres, lo anterior se refleja en el entendimiento y aplicación de estos conocimientos en la práctica.
- En cuanto al uso de aplicativos de software libre en general se concluye que es bajo en el ámbito académico, pero al ser enfocado en áreas específicas de ingeniería su preferencia es alta, factores como la utilidad, aplicación en algunas fases de ingeniería de software como análisis y diseño, el ser gratuito y flexibilidad lo convierte en una buena alternativa al software propietario.
- Por último, el aprendizaje de software libre enfocado directamente al aprendizaje de UML tiene una aceptación muy alta y su inclusión es bien recibida, pese a que aún se tiene una confianza algo baja de algunos sectores educativos.

Se puede concluir a nivel general que el uso de aplicativos de software libre en la enseñanza de ingeniería de software es pertinente y por tanto genera muy buenos resultados.

9. RECOMENDACIONES

- Para la implementación de software libre en la enseñanza se debe tener una infraestructura de computo adecuada y actualizada, lo anterior con el fin de poder instalar y ejecutar los sistemas software de forma correcta.
- Se debe validar el software más indicado para el desarrollo de la asignatura de ingeniería de software dependiendo de las necesidades del curso.
- Implementar y orientar técnicas de enseñanza dirigidas al aprendizaje de ingeniería de software a partir de software libre, en cuanto se identifiquen las técnicas, ejecutarlas en el curso para validar el aprendizaje y su efectividad.
- Validar con la institución de educación superior si la implementación de software libre está autorizada, y si esta no afecta comprar de software dedicado.
- Se debe tener en cuenta que los cursos evaluados varían de conocimientos previos por lo que es necesario evaluar niveles de apropiación de software libre en el comienzo del curso.
- El trabajo se orientó a una institución en específico, estos resultados pueden variar según el núcleo de enseñanza y la posición de la institución de educación superior frente al software libre.

10. BIBLIOGRAFÍA

- Abello, D., Hederich, C., & Hernández, C. (2010). Caracterización de los estilos de enseñanza de una muestra de docentes de la Universidad Pedagógica Nacional. Bogotá: UPN.
- Adell, J., & Bernabé, I. (2007). Software libre en educación. Madrid: McGraw Hill.
- Barreal, J., & Ugolini, A. (2014). El software libre como recurso didáctico en la enseñanza de métodos cuantitativos en los grados de economía y empresas. Research Gate.
- Camargo, A. (2010). Dimensiones interactiva, discursiva y didáctica del estilo de enseñanza, el caso de las ciencias naturales. Bogotá: UPN.
- Chavarría, J. (2005). Software libre, la alternativa tecnológica para la educación. Madrid: Actualidades.
- Concejo de Bogotá. (2006). Acuerdo 331 de 2006 Concejo de Bogotá D.C. Bogotá.
- Cuervo, M. (2005). La ingeniería de software libre aplicada a los proyectos informáticos. Reportes técnicos en ingeniería del software, 30-35.
- Difference Between. (14 de noviembre de 2017). Difference between UML 1.0 and UML 2.0. Obtenido de <http://www.differencebetween.info/difference-between-uml-1-and-uml-2>
- Fonseca, O. (2013). Efecto de los estilos de enseñanza y aprendizaje sobre el rendimiento académico. México: Universidad Baja California.
- Fonseca, O. (2013). Relación entre los estilos de enseñanza de la Maestría en Didáctica de las Ciencias y la Formación en Pregrado. Bogotá: UAC.
- Frasser, W. (2004). Historia y evolución de la ingeniería de software. Bogotá: Universidad Sergio Arboleda.
- GNU Operating System. (2017). Free Software for Education. Obtenido de <https://www.gnu.org/software/free-software-for-education.en.html>
- Gómez, M., & Polanía, R. (2008). Estilos de enseñanza y modelos pedagógicos. Bogotá: Universidad de La Salle.
- Hernández, E. (2012). El Lenguaje Unificado de Modelado (UML). Valencia: UPV.
- Hernández, S., & Cuevas, J. (2013). Programas informáticos de uso libre y su aplicación en la enseñanza de la estadística. Revista de investigación operacional, 166-174.

Jaeger, W. (2000). Paidea: los ideales de la cultura griega. En W. Jaeger, Paidea: los ideales de la cultura griega (pág. 19). México: Fondo de Cultura Económica.

MEN. (13 de 11 de 2009). Ministerio de Educación Nacional. Recuperado el 10 de noviembre de 2018, de El software libre busca adeptos: <https://www.mineduacion.gov.co/cvn/1665/w3-article-195726.html>

MEN. (2018). Ministerio de Educación Nacional. Recuperado el 10 de noviembre de 2018, de Buscan que el software sea libre: <https://www.mineduacion.gov.co/cvn/1665/article-165146.html>

Orjuela, H., & Hurtado, A. (2010). Perfeccionamiento de un nuevo simulador interactivo, bajo software libre gnu/linux, como desarrollo de una nueva herramienta en la enseñanza y aprendizaje de la física. *Latin-American Journal of Physics Education*, 200-203.

República de Colombia. (1982). Ley 23 de 1982. En: Congreso de la república.

Rendón, A. (2010). Los estilos de enseñanza en la Universidad de Antioquia. Medellín: Universidad de Antioquia.

Rojas, C., Díaz, C., Vergara, J., Alarcón, P., & Ortiz, M. (septiembre-diciembre de 2016). Teaching and Learning Styles in Higher Education: Analysis of Student Teachers' Preferences in an English Pedagogy Program at Three Chilean Universities. *Educare*, 20(3), 250-378.

Rueda, F. (2004). Historia y actualidad del software libre. Obtenido de <https://mmujica.files.wordpress.com>

Stallman, R. (2016). Richard Stallman's Personal Site. Recuperado el 11 de noviembre de 2018, de <http://www.stallman.org/>

Universidad de castilla. (2015). Licencias para software libre. En: Biblioteca universitaria.

Vigotsky, J. (1978). *Mind in Society: The development*. Cambridge: Harvard University Press.

Virvou, M., & Mondridou, M. (2001). Agregando un componente modelador e instructor a la arquitectura de las herramientas de autor ITS. *International Journal of Artificial Intelligence in Education*, 12(2), 185-201.

Zaragaza, J., & Noguerras, J. (2008). *Introducción a la ingeniería de software*. Zaragoza: Universidad de Zaragoza.

Zhang, L. (2009). From conceptions of effective teachers to styles of teaching: Implications for higher education. *Larning & Individual Differences*, 19(1), 113-118.

ANEXOS

Encuesta:

Nombre: _____ Sexo: M__ F__

Edad: 16-20__ 21-25__ 26-30__ +30__

Hora Inicio: _____ Hora Fin: _____

Instrucciones: Seleccione un valor de la escala de 1 a 5, donde 1 significa completamente en desacuerdo y 5 que está completamente de acuerdo

1. Entiendo el concepto de ingeniería de software

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

2. Entiendo el concepto de software libre

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

3. Uso software libre frecuentemente

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

4. En la asignatura de ingeniería de software o programación entendí la aplicabilidad de UML

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

5. Utilicé algún software en el diseño de UML

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

6. Está de acuerdo con el uso de software libre en la enseñanza

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

7. Señale algunas ventajas que conozca del software libre

___ Es gratuito

___ Existe gran comunidad de desarrollo

___ Es fácil de aprender

___ Puede distribuirlo libremente

___ Es acomodado al negocio

___ Útil al aprendizaje de UML

8. Considera importante y útil que se utilice software libre para la enseñanza de UML

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

9. Considera su formación adecuada con el uso de software libre

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

10. Sugeriría a su institución educativa implementar la enseñanza de ingeniería de software con la utilización de software libre

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

11. Señale si conoce alguna de estas herramientas

___ ArgoUML

___ Visual Paradigm Community Edition

___ StarUML

___ UMLET

___ BOUML

___ Umbrello

___ Draw.io

___ No conozco ninguna

12. Considera importante la inclusión de software libre en la enseñanza de ingeniería de software

___ SI

___ NO

13. Su nivel de formación se vería afectado si se enseñara con software libre

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

14. En algún momento ha utilizado software libre en el diseño de diagramas UML

___ SI

___ NO

15. Del siguiente listado considera interesante del software libre para su utilización.

___ Formación

___ Enseñanza en la Instalación de software

___ Implicación institucional

___ Ahorro de recursos

16. Preferiría utilizar aplicativos de software libre en su aprendizaje que software propietario

Muy en desacuerdo

Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

17. Estaría de acuerdo en que su empleo se utilizara únicamente software libre
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

18. ¿Está de acuerdo con las políticas de su institución educativo en cuanto al uso de software libre?
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

19. Desde su punto de vista, el nivel de equipos de cómputo de la institución permitiría la inclusión de software libre
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

20. El nivel de apropiación de sus docentes en la enseñanza de software libre es el adecuado.
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

21. ¿Encuentra útil el uso de software libre en la enseñanza de ingeniería de software?
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

22. Siente que requiere más conocimiento del uso de software libre
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

23. ¿Ha utilizado algún software libre para la recolección de requisitos en la fase de análisis en ingeniería de software?
___SI ___NO

24. ¿Ha utilizado algún software libre para realizar diagrama de clase en la fase de diseño en ingeniería de software?
___SI ___NO

25. Se ha sentido a gusto usando software libre
Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

26. Que tanto recomendaría el uso de software libre en su aprendizaje

Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

27. Cual es nivel de confianza para su proceso aprendizaje si únicamente se utiliza software libre en el aprendizaje de UML

Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

28. Cuál es su opinión respecto a si debería mezclarse la enseñanza de ingeniería de software usando software libre y software privativo

Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

29. Que tan dispuesto estaría en usar otro sistema operativo que no sea Microsoft Windows si de eso dependiera el uso de software libre en su proceso de enseñanza de ingeniería de software

Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---

30. Se sentiría satisfecho con el hecho que el software libre en algunas ocasiones tiene menos herramientas que un software privativo pero que de igual manera contiene lo necesario para el diseño de diagramas UML en fase de diseño

Muy en desacuerdo Muy de acuerdo

1	2	3	4	5
---	---	---	---	---